GEORGIA INSTITUTE OF TECHNOLOGY
OFFICE OF RESEARCH ADMINISTRATION

RESEARCH PROJECT INITIATION

Postal Gr Cd

Date: June 6, 1975

Project Title: A Study of Turbulent Flows About Oscillating Airfoils

Project No: E-16-669

Gr Cd Army

Principal Investigator: Dr. James C. Wu

Sponsor: U. S. Army Research Office; Research Triangle Park, N.C.

Agreement Period: From June 1, 1975 Until ~~May 31, 1977~~ 8/31/78

Type Agreement: Grant No. ~~DAHC04~~ DAAG29-75-G-0147

Amount: $50,000 ARO
9,202 GIT Cost Sharing (E-16-359)
$59,202 Total

Reports Required:
Progress Report (1st 9/30/75 next 3/31/75)
Technical (When justified)
Final

Sponsor Contact Person (s):

Technical
ARO Engineering Division
Box CM, Duke Station
Durham, North Carolina 27706

Administrative
thru ONR
Mr. R. J. Whitcomb
ONR RR
Campus

Assigned to: ~~Aerospace Engineering~~

COPIES TO:

Principal Investigator	Library
School Director	Rich Electronic Computer Center
Dean of the College	Photographic Laboratory
Director, Research Administration	Project File
Director, Financial Affairs (2)	
Security-Reports-Property Office ✓	
Patent Coordinator	Other

RA-3 (6-71)

GEORGIA INSTITUTE OF TECHNOLOGY
OFFICE OF CONTRACT ADMINISTRATION

SPONSORED PROJECT TERMINATION

Date: May 3, 1979

Project Title: A Study of Turbulent Flows About Oscillating Airfoils

Project No: E-16-669

Project Director: Dr. James C. Wu

Sponsor: U. S. Army Research Office

Effective Termination Date: 8/31/78

Clearance of Accounting Charges: 8/31/78

Grant/Contract Closeout Actions Remaining:

_ Final Invoice and Closing Documents
_ Final Fiscal Report
_ Final Report of Inventions
_ Govt. Property Inventory & Related Certificate ← Submitted 16 Feb 79
_ Classified Material Certificate
X Other Please send copy of Final Report for our records.

Assigned to: Aerospace Engineering (School/Laboratory)

COPIES TO:

Project Director
Division Chief (EES)
School/Laboratory Director
Dean/Director—EES
Accounting Office
Procurement Office
✓Security Coordinator (OCA)
Reports Coordinator (OCA)
Library, Technical Reports Section
Office of Computing Services
Director, Physical Plant
EES Information Office
Project File (OCA)
Project Code (GTRI)
Other

CA—4 (3/76)

PROGRESS REPORT NO. 1

1. ARO PROPOSAL NUMBER: JBG/G 4002.74

2. PERIOD COVERED BY REPORT: June 1, 1975 to September 30, 1975

3. TITLE OF PROPOSAL: A Study of Turbulent Flows About Oscillating Airfoils

4. GRANT NUMBER: DAHC04-75-G-0147 (Project No. 13100 - E)

5. NAME OF INSTITUTION: Georgia Institute of Technology

6. AUTHOR OF REPORT: James C. Wu

7. LIST OF MANUSCRIPTS SUBMITTED OR PUBLISHED UNDER ARO SPONSORSHIP DURING THIS PERIOD, INCLUDING JOURNAL REFERENCES:

   None

8. SCIENTIFIC PERSONNEL SUPPORTED BY THIS PROJECT AND DEGREES AWARDED DURING THIS REPORTING PERIOD:

   Dr. James C. Wu, Professor

   R. Sugavanam, Graduate Student

   M. M. Wahbah, Graduate Student

BRIEF OUTLINE OF RESEARCH FINDINGS

During the reporting period, the time-dependent separated turbulent flow problem was formulated as integro-differential equations on the basis of a two-equation model. The formulation uses the velocity vector, the vorticity vectors, the turbulence energy and turbulence energy dissipation as dependent field variables and partitions the problem into its kinetic and kinematic aspects. The kinetic aspect consists of differential transport equations describing the change of vorticity, turbulence energy, and turbulent energy dissipation through diffusion, convection, production and dissipation processes. The kinematic aspect relates the vorticity distribution and the velocity boundary condition at a given instant of time to the velocity distribution at that instant. An integral representation for the velocity vector is used in place of the familiar differential continuity and vorticity definition equations in the kinematic part of the problem.

The theoretical basis of the integro-differential method was developed previously for laminar flows. The integro-differential formulation for turbulent flows was found to retain many important features of the laminar flows. The kinematics of the mean turbulent motion was found to be identical to that for laminar motion so that the integral representation for the velocity vector developed for the laminar flows goes over to turbulent flows in a straight-forward manner. The integral representation permits the velocity to be computed explicitly, point by point. This unique ability led to a number of advantages: (1) the solution field can be confined to the viscous region of the flow, (2) the confined solution field can be segmented into compartments and the computation of field variables interior of each compartment can be accomplished independently of the other compartments, (3) the vorticity

boundary condition on solid surfaces needed for the kinetic part of computation can be established correctly through kinematic considerations, (4) for external flows, the problem of satisfying the velocity boundary conditions infinitely away from the solid is eliminated.

The momentum equation for the mean turbulent flow contains a Reynolds stress term in addition to the terms present in the laminar equation. Consequently, the mean vorticity transport equation for turbulent flows contains additional terms representing the convection and production of vorticity by turbulent rate of strain. In addition to the mean vorticity transport equation, two additional time-dependent transport equations for the turbulence energy and the energy dissipation are established. The set of three time-dependent transport equations are to be solved along with the integral representation using empirical constants that were well established for steady flows.

The new formulation was calibrated by solving the problem of time-dependent turbulent Couette flow formation. Using boundary values of the vorticity, the turbulence energy, and turbulence energy dissipation anticipated for steady state Couette flow, a time-dependent solution was obtained. The mean velocity profiles obtained was found to approach the steady flow experimental data of Reichardt asymptotically in the limit of large time.

During the next twelve months of this research, the problem of determining the time-dependent boundary conditions for the vorticity, the turbulence energy, and the turbulence energy dissipation, shall be studied critically. These problems associated with time-dependent boundary conditions for turbulent flow are among the least understood aspects of the present project and are expected to require considerable amounts of effort for their resolution.

PROGRESS REPORT

20 COPIES REQUIRED

1. ARO PROPOSAL NUMBER: JBG/G 4002.74

2. PERIOD COVERED BY REPORT: October 1, 1975 to June 30, 1976

3. TITLE OF PROPOSAL: A Study of Turbulent Flows About Oscillating Airfoils

4. CONTRACT OR GRANT NUMBER: DAHC04-75-G-0147 (Project No. 13100 - E)

5. NAME OF INSTITUTION: Georgia Institute of Technology

6. AUTHOR(S) OF REPORT: James C. Wu

7. LIST OF MANUSCRIPTS SUBMITTED OR PUBLISHED UNDER ARO SPONSORSHIP DURING THIS PERIOD, INCLUDING JOURNAL REFERENCES:

"A Numerical Study of Viscous Flow Around an Airfoil," by James C. Wu and S. Sampath, AIAA Paper 76-337, (Pre-print attached)

8. SCIENTIFIC PERSONNEL SUPPORTED BY THIS PROJECT AND DEGREES AWARDED DURING THIS REPORTING PERIOD:

Dr. James C. Wu, Professor

R. Sugavanam, Graduate Student

M. M. Wahbah, Post-Doctoral Fellow

13100E

DR. JAMES C. WU
GEORGIA INSTITUTE OF TECHNOLOGY
SCHOOL OF AEROSPACE ENGINEERING
ATLANTA, GA 30332

BRIEF OUTLINE OF RESEARCH FINDINGS

During the reporting period, the formulation of integro-differential equations for time-dependent turbulent flows has continued. This formulation partitions the problem into its kinetic and kinematic aspects. The kinematic aspect relates the mean vorticity distribution at any instant of time to the mean velocity distribution at that instant. This relation is expressed as an integral representation for the velocity vector and permits an explicit, point by point, computation of the velocity field. The kinetic aspect consists of differential transport equations describing the change of vorticity, turbulence energy, and turbulence energy dissipation through diffusion, advection, production, and dissipation. A turbulent coefficient of diffusivity is modeled in terms of the turbulence energy and the turbulence energy dissipation.

Two relatively simple time-dependent one-dimensional turbulent flow problems have been studied as a part of an effort to calibrate the formulation. The first of these problems is the formulation of Couette flow between two parallel plates which are set into motion in opposite directions in their own planes impulsively. The vorticity is initially concentrated at the two plate surfaces. At subsequent times the vorticity spreads into the fluid domain by turbulent diffusion. Transient solutions were obtained. The solution was shown to asymptotically approach the steady-state experimental data of Reichardt with increasing time and to agree with the latter excellently in the limit.

The second problem treated was that of the flow induced by an infinite plate sinusoidally oscillating in its own plane. The results obtained for the wall shear stress are in good qualitative agreement with those obtained by Nash, Carr, and Singleton (AIAA Journal Vol. 13, pp. 167-173) for the flow past an oscillating finite flat plate.

The above problems involve only one space dimension. The transport equations for the two problems are solved using Crank-Nicolson finite-difference and explicit finite-difference methods respectively. Currently, the two-dimensional transient problem of a finite flat plate set into motion impulsively in its own plane is being studied. Several promising finite-difference methods are being investigated in connection with this study. It is found that the use of the law of the wall in conjunction with the integral representation yields reasonable velocity and vorticity profiles.

PROGRESS REPORT

1. ARO PROPOSAL NUMBER: JBG/G 4002.74

2. PERIOD COVERED BY REPORT: July 1, 1976 to December 31,11976

3. TITLE OF PROPOSAL: A Study of Turbulent Flows About Oscillating Airfoils

4. CONTRACT OR GRANT NUMBER: DAHC04-75-G-0147 (Project No. 13100 - E)

5. NAME OF INSTITUTION: Georgia Institute of Technology

6. AUTHOR(S) OF REPORT: James C. Wu

7. LIST OF MANUSCRIPTS SUBMITTED OR PUBLISHED UNDER ARO SPONSORSHIP DURING THIS PERIOD, INCLUDING JOURNAL REFERENCES:

"A Numerical Study of Viscous Flow Around an Airfoil," by James C. Wu and S. Sampath, AIAA Paper 76-337, July, 1976.

(Pre-prints submitted with previous progress report.)

8. SCIENTIFIC PERSONNEL SUPPORTED BY THIS PROJECT AND DEGREES AWARDED DURING THIS REPORTING PERIOD:

Dr. James C. Wu, Professor

R. Sugavanam, Graduate Student

M. M. Wahbah, Post-Doctoral Fellow

13100E

DR. JAMES C. WU
GEORGIA INSTITUTE OF TECHNOLOGY
SCHOOL OF AEROSPACE ENGINEERING
ATLANTA, GA 30332

BRIEF OUTLINE OF RESEARCH FINDINGS

The development of an integro-differential formulation for time-dependent turbulent flows utilizing a two-equation turbulence model has been completed. Several pitfalls which were not at first obvious have been removed. A detailed description of the formulation, which contains a kinematic equation expressed as an integral representation for either the velocity vector or the stream function and three kinetic equations expressed in differential forms for the vorticity, the turbulence energy, and the turbulence energy dissipation, has been included in a technical report being finalized. It has been found that the integro-differential formulation for turbulent flows retains the distinguishing feature of the formulation for linear flows. That is, the integro-differential approach permits the solution field to be confined to the viscous region and therefore offers substantial computational advantages for turbulent flows, as it does for laminar flows. This ability to confine the solution field is obviously of great importance in the numerical solution of problems involving significant regions of flow separation.

With a traditional finite-difference or finite-element method, the major portion of the required computation is spent in treating the kinematic equation. The integro-differential approach so drastically reduces the kinematic computation that the treatment of the kinematic equation now represents the major portion of the needed computation. The solution of the kinetic equations therefore replaces the solution of the kinematic equation as the limiting factor on computational speed for the overall problem. This limitation is more acute for turbulent flows than it is for laminar flows since the former involves three kinetic equations while the latter involves only one. In order to minimize the computation necessary for the kinetic equations, a hybrid finite difference-finite element, implicit-explicit method is being examined. For laminar flows, this method has been shown, under a separate research project, to possess the advantages and to alleviate the disadvantages of the various methods that compose the hybrid approach. The very substantial reduction in the kinetic computation offered by the hybrid method for laminar flows is expected to be retained for turbulent flows.

Prior to the reporting period, two relatively simple turbulent flow problems involving one space dimensions - the Couette flow formation between two infinite flat plates and the flow induced by an oscillating infinite plate - were treated in order to calibrate the integro-differential approach. During the reporting period, more complicated problems involving two space dimensions have been studied. Numerical solutions have been obtained for the problem of a finite flat plate set into motion impulsively in its own plane, at a Reynolds number of 5.3×10^7 based on plate length. Steady state results, in the limit of large time, for the velocity, skin friction, and turbulence energy has been found to agree well with available experimental data. A computer program has been prepared for the numerical solution of the problem of turbulent flow past a circular cylinder set into motion impulsively in a direction perpendicular to its axis.

PROGRESS REPORT

1. ARO PROPOSAL NUMBER: JBG/G 4002.74

2. PERIOD COVERED BY REPORT: January 1, 1977 to June 30, 1977

3. TITLE OF PROPOSAL: A Study of Turbulent Flows About Oscillating Airfoils

4. CONTRACT OR GRANT NUMBER: DAHCO4-75-G-0147 (Project No. P-13100 - E)

5. NAME OF INSTITUTION: Georgia Institute of Technology

6. AUTHOR(S) OF REPORT: James C. Wu

7. LIST OF MANUSCRIPTS SUBMITTED OR PUBLISHED UNDER ARO SPONSORSHIP DURING THIS PERIOD, INCLUDING JOURNAL REFERENCES:

a. "A Method for the Numerical Solution of Turbulent Flow Problems," by J. C. Wu and S. Sugavanam, AIAA Paper No. 77-649, Proceedings' of the 3rd Computational Fluid Dynamics Conference, pp. 168-177, 1977.

b. "Some Numerical Solutions of Turbulent Flow Problems by the Use of Integral Representation," by J. C. Wu, M. M. Wahbah and A. Sugavanam, Proceedings of the Symposium on Application of Computer Methods in Engineering, in print, 1977.

8. SCIENTIFIC PERSONNEL SUPPORTED BY THIS PROJECT AND DEGREES AWARDED DURING THIS REPORTING PERIOD:

Dr. James C. Wu, Professor

R. Sugavanam, Graduate Student

M. M. Wahbah, Post-Doctoral Fellow

13100E

R. JAMES C. WU
EORGIA INSTITUTE OF TECHNOLOGY
CHOOL OF AEROSPACE ENGINEERING
TLANTA, GA 30332

BRIEF OUTLINE OF RESEARCH FINDINGS

During the reporting period, considerable progress has been made in the development of numerical methods for both the time-dependent and the steady turbulent flow problems. Several problems have been solved numerically using the $k-\varepsilon$ (turbulent kinetic energy and energy dissipation) model for turbulence. An integro-differential formulation has been used in the time-dependent flow case and an integral-representation formulation has been developed in the steady flow case to confine the solution field to the vortical region of the flow. This distinguishing feature of the integro-differential and integral-representation formulations, which led to remarkable computational efficiency for laminar separated flow computations, has been incorporated into a computer code for the turbulent flow.

With either formulations, the vorticity field is utilized to partition the overall problem into its kinematic and kinetic aspects. The kinematic aspect deals with the relations between the instantaneous vorticity and velocity, or the vorticity and stream function, fields, and is formulated as an integral representation of the velocity, or the stream function, field. The kinetic aspect deals with the transport of vorticity and is expressed as a differential vorticity transport equation for the time-dependent flow. For the steady flow, the steady vorticity transport equation is recast into an integral representation for the vorticity field. These aspects are supplemented by differential transport equations for k and ε. A knowledge of k and ε fields is utilized in the computation of the Reynolds stress through the Boussinesq concept.

Results were obtained for several time-dependent problems. They include flows past a finite flat plate with either a steady or an oscillating main-stream and flows past a circular cylinder. In each problem, the motion of the fluid was considered to result from an impulsively started motion of the solid body. The Reynolds numbers based on the freestream velocity and the dimension of the solid body are of the order of 10^6. Specific Reynolds numbers were selected in order to compare the numerical results obtained in this study with experimental data available in the literature. For the flat plate problems, the time-dependent solutions were carried to a sufficiently large time level so that either a steady-state or a periodic condition was reached. Comparisons between the numerical results and experimental data were made for profiles of mean velocity, turbulent kinetic energy, and Reynolds stress on the plate and in the wake downstream of the plate. Very good agreement was observed in each case. The computation of the flow around the circular cylinder is still in progress. Flow patterns and skin friction distributions on the cylinder have been computed up to a dimensionless time level of two, which corresponds to a movement of the cylinder relative to the freestream through a distance of two radii. Steady flow was not reached at this time level. Nevertheless, the computed separation point is in good agreement with experimental data. The computed skin friction shows the same pattern as that experimentally measured.

The basic computation loop to advance the time-dependent solution from an old time level, at which the vorticity, velocity, and Reynolds stress fields are known, to a new time level includes (1) the kinetic computation of a new vorticity field, (2) the kinematic computation of a corresponding new velocity field, (3) the computation of k and ε, which permits the new Reynolds stress field to be established. During the reporting period, a Fourier series method has been developed for the kinematic aspect of the problem. This method provides a highly accurate and fast means of computing the velocity field. With a traditional finite-difference or finite-element method, the major portion of the required computation is spent in treating the kinematic equation. The integral representation of the kinematics of the problem has so drastically reduced the kinematic computation that the kinetic and turbulence modelling computations now represents the major portion of the needed computation. The solution of the transport equations for k, ε, and vorticity therefore replaces the solution of the kinematic equation as the limiting factor on computation speed for the overall problem. Considerable efforts have been spent during the reporting period to develop fast and accurate methods for solving the transport equations.

For steady flows, the basic computation loop described above is used in an iterative procedure to advance the solution from an old iteration to a new one. It has been found that for steady flows, the kinetic equation, i.e., the steady vorticity transport equation, can be recast into an integral representation for the vorticity field. A very efficient and general computer program based on this complete integral formulation has been previously prepared for laminar internal flows. The k-ε turbulence model is being incorporated into this existing program. Some results have been obtained for the turbulent channel flow problem.

ABSTRACTS

A METHOD FOR THE NUMERICAL SOLUTION OF TURBULENT FLOW PROBLEMS

by J. C. Wu and A. Sugavanam

Paper appears as AIAA Paper No. 77-649, Proceedings of the 3rd AIAA Computational Fluid Dynamic Conference, 1977.

An integro-differential formulation, previously established for the study of incompressible laminar flow problems, is modified and used in conjunction with the turbulent kinetic energy-energy dissipation rate model for the study of turbulent flow problems. It is demonstrated that the kinematics of the mean turbulent motion is identical to that of the laminar motion. The kinetic aspect of the turbulent flow is significantly more complex. The integro-differential formulation for the turbulent flow retains the important features of the laminar flow and permits the solution field to be confined to the vortical region. Numerical results are presented for several problems involving simple flow boundaries and compared with available experimental data.

SOME NUMERICAL SOLUTIONS OF TURBULENT FLOW PROBLEMS BY THE USE OF INTEGRAL REPRESENTATIONS

by J. C. Wu, S. Sugavanam, and M. Wahbah

Paper to appear in Proceedings of the Symposium on Applications of Computer Methods in Engineering, 1977.

The use of integral representations in the solution of turbulent flow problems is discussed. It is shown that numerical procedures previously established for steady and time-dependent laminar flows go over in a straightforward manner to turbulent flows. The important advantages previously demonstrated for laminar flows are retained in the current study of turbulent flows. In particular, the integral-representation approach permits the solution field to be confined to the vortical region and offers superior computational efficiency and solution accuracy.

PROGRESS REPORT

(TWENTY COPIES REQUIRED)

1. ARO PROPOSAL NUMBER: JBG/G 4002.74

2. PERIOD COVERED BY REPORT: July 1, 1977 to December 31, 1977

3. TITLE OF PROPOSAL: A Study of Turbulent Flows About Oscillating Airfoils

4. CONTRACT OR GRANT NUMBER: DAHC04-75-G-0147 (Project No. P-13100-E)

5. NAME OF INSTITUTION: Georgia Institute of Technology

6. AUTHOR(S) OF REPORT: James C. Wu

7. LIST OF MANUSCRIPTS SUBMITTED OR PUBLISHED UNDER ARO SPONSORSHIP DURING THIS PERIOD, INCLUDING JOURNAL REFERENCES:

"Some Numerical Solutions of Turbulent Flow Problems by the Use of Integral Representation," by J.C. Wu, M.M. Wahbah and A. Sugavanam, Proceedings of the Symposium on Application of Computer Methods in Engineering, pp. 983-992, August, 1977.

8. SCIENTIFIC PERSONNEL SUPPORTED BY THIS PROJECT AND DEGREES AWARDED DURING THIS REPORTING PERIOD:

Dr. James C. Wu, Professor

R. Sugavanam, Graduate Student

M. M. Wahbah, Post-Doctoral Fellow

13100-E

Dr. James C. Wu
Georgia Institute of Technology
School of Aerospace Engineering
Atlanta, GA 30332

BRIEF OUTLINE OF RESEARCH FINDINGS

During the reporting period, the k -ε (turbulent kinetic energy and energy dissipation) model for turbulence has been used together with the integro-differential method to treat turbulent separated flow problems. Results were obtained for a flow past a circular cylinder at a Reynolds number based on the cylinder diameter of 3.6×10^6. The cylinder is set into motion impulsively at a certain instant of time and thereafter kept in motion at a uniform velocity. The complicated time-dependent flow after the impulsive start is computed. The computation was carried up to a dimensionless time level of about seven, that is, a time level corresponding to the movement of the cylinder relative to the freestream by a distance of seven cylinder diameters. At this large time level, the time rate of change of the flow was no longer significant. The numerical results obtained at this large time level were found to be in reasonable qualitative agreement with the experimental data of Achenbach for the same Reynolds number. In view of the substantial scattering of experimental data for this type of complicated high Reynolds number flows, the observed agreement was encouraging. As reported previously (see "A Method for the Numerical Solution of Turbulent Flow Problems", J.C. Wu and S. Sugavanam, Proceedings of the 3rd AIAA Computational Fluid Dynamics Conference, pp. 168-177, 1977), for simpler types of flow where definitive experimental data existed, the present computation procedure yielded numerical results in excellent quantitative agreement with the experiments. The results obtained during this reporting period indicate that the present procedure is suitable for computing complex separated flows. Accordingly, efforts have been initiated to generalize the present procedure for application to the problem of time-dependent turbulent flows around airfoil.

Two obstacles were encountered in obtaining numerical solutions for the circular cylinder problem. They were (1) the relatively large amounts of storage and computer time required and (2) the tendency for the numerical solution to become unstable as time progresses. The second obstacle was removed by simulating the source and dissipation terms in the transport equations for the vorticity, the turbulent kinetic energy, and the energy dissipation at the new time level in each computation loop to advance the solution in time. The first obstacle was partly a consequence of the need to solve additional transport equations (for k and ε) when the flow is turbulent. As described in previous reports, the integro-differential method that has been developed by this research group is highly efficient in comparison to other numerical methods. For external laminar flows, existing computer programs constructed on the basis of the integro-differential approach require about 10 minutes of CDC 6600 CPU time to advance the numerical solution by one dimensionless unit of physical time, i.e., the time interval during which the airfoil advances by one chord length relative to the freestream. Consequently, drastic further improvements in solution efficiency is no longer a critical factor in the development of a general-purpose user-oriented package of computer code for external laminar flows in two-dimensions. The numerical solution of the equations for k and ε , however, places additional requirements on the allowable grid spacing

and time intervals. These requirements, coupled with the need to test and calibrate the available turbulence models extensively, made it important to develop numerical methods that are even more efficient than the present versions of the integro-differential method. To provide a means of extensively testing and calibrating various turbulence models, a standardize package of computer code was recently prepared by Dr. M.M. Wahbah for steady internal flows using an integral representations approach (see "Numerical Solution of Viscous Flow Equations Using Integral Representations," Lecture Series in Physics, Vol. 59, pp. 448-453, Springer-Verlag, 1976.). This package is user-oriented and highly efficient. It is being utilized in calibrating various turbulence models in the steady state limit. In addition to this standardized package, a Fourier series technique was developed and used to improve drastically the efficiency of the kinematic part of the computation. This technique was successfully used to treat the circular cylinder problem. For the airfoil problem, a numerical method was developed to transform conformally the airfoil boundary to a circle prior to the use of the Fourier series technique.

E16-669

PROGRESS REPORT

(TWENTY COPIES REQUIRED)

1. ARO PROPOSAL NUMBER: JBG/G 4002.74

2. PERIOD COVERED BY REPORT: January 1, 1978 to June 30, 1978

3. TITLE OF PROPOSAL: A Study of Turbulent Flows About Oscillating Airfoils

4. CONTRACT OR GRANT NUMBER: DAAG29-75-G-014 (Project No. P-13100-E)

5. NAME OF INSTITUTION: Georgia Institute of Technology

6. AUTHOR(S) OF REPORT: James C. Wu

7. LIST OF MANUSCRIPTS SUBMITTED OR PUBLISHED UNDER ARO SPONSORSHIP DURING THIS PERIOD, INCLUDING JOURNAL REFERENCES:

   See attached list

8. SCIENTIFIC PERSONNEL SUPPORTED BY THIS PROJECT AND DEGREES AWARDED DURING THIS REPORTING PERIOD:

   Dr. James C. Wu, Professor

   A. Sugavanam, Graduate Student

   M. M. Wahbah, Post-Doctoral Fellow

13100-E

Dr. James C. Wu
Georgia Institute of Technology
School of Aerospace Engineering
Atlanta, GA 30332

BRIEF OUTLINE OF RESEARCH FINDINGS

A major objective of this research program is to develop a general computational capability for the prediction of time-dependent turbulent flows which may involve appreciable flow separation and strong viscous-inviscid interaction . Major obstacles to this development are: (1) the excessive computer time and data storage needs for the solution of problems involving complex flow boundaries, and (2) the lack of accurate and general methods of describing turbulent transport phenomena in separated flow regions.

Towards the above stated objectives, a numerical method utilizing an integro-differential formulation of the problem has been studied. This new formulation possesses a number of highly advantageous attributes, the most important one being an ability to confine the solution field to the vortical region of the flow. To utilize these attributes, however, entirely new numerical procedures needed to be established. During the reporting period, the development of the integro-differential method has reached a reasonably stage of completion. By employing various techniques specially developed for the new formulation, such as coordinate transformation, hybrid finite element-finite difference method, Fourier approximation and flowfield segmentation, user-oriented computer programs that are useful under fairly general circumstances have been constructed. These programs have been used to obtain numerical solutions for many complex laminar flows, including several time-dependent flows past airfoil sections. The computer time used are not excessive even with relatively slow computers such as the CDC-6400. The first obstacle stated above has been removed for two-dimensional flows.

Because of their necessary empirical foundation, extensive testing and calibration are necessary for establishing the validity and the range of application of proposed turbulent models. The remarkable solution efficiency of the present method made it possible to carry out this task efficiently and relatively economically. Several algebraic and differential models have been examined critically, under various flow circumstances. Encouraging numerical results have been obtained for many types of flows. These results suggested several possible improvements and refinements of the models. The two aspects of the study described above -- numerics and turbulence modeling -- has been utilized to study flows past airfoils.

The number of manuscripts prepared during the reporting period, as indicated on the attached list, is large for the size of the present project. The preparation of these manuscripts indeed somewhat taxed the energy of the present investigators during the reporting period. It was felt,however, that significant progress has been made recently in many aspects of the present research. It is appropriate to document the information generated and to make it available in the open literature in a timely manner. Considerable interest in the present work has been shown by many researchers in the fluid dynamics community. A large number of inquiries and requests for information have been received by the principal investigator in recent months.

LIST OF MANUSCRIPTS

a. "A Method for the Numerical Solution of Turbulent Flow Problems", by J. C. Wu and A. Sugavanam; revised manuscript submitted April, 1978; scheduled for AIAA Journal, Vol. 16, No. 9, 1978.

b. "Integral-Representation Approach for Time-Dependent Viscous Flows", by J. C. Wu and Y. M. Rizk; manuscript submitted June, 1978; presented at the Sixth International Conference on Numerical Methods in Fluid Dynamics, June, 1978; in print by Springer-Verlag as a part of Lecture Notes in Physics, 1978.

c. "Numerical Solution of Unsteady Flow Problems Using Integro-Differential Approach", by J. C. Wu, M. M. Wahbah, and A. Sugavanam; manuscript submitted in April, 1978; schedule for presentation at the Symposium on Nonsteady Fluid Dynamics of the ASME 1978 Winter Annual Meeting in December of 1978; in print, Symposium on Nonsteady Fluid Dynamics Proceedings.

d. "Computation of Internal Flows with Arbitrary Boundaries Using the Integral Representations Method", by M. M. Wahbah, Georgia Institute of Technology Report, March, 1978.

e. "A Rarefied Gas Dynamics Numerical Method Applied to Problems in Statistical Turbulence", by R. Srinivasan, D. P. Giddens, L. H. Bangert, and J. C. Wu, manuscript submitted June, 1978, for publication in the Proceedings of the Eleventh International Symposium on Rarefied Gas Dynamics.

f. "Prospects for Computational Aerodynamics", by J. C. Wu, published in NASA CP-2032, pp. 221-227, February, 1978.

ABSTRACTS

A METHOD FOR THE NUMERICAL SOLUTION OF TURBULENT FLOW PROBLEMS

by J. C. Wu and A. Sugavanam

To appear in Vol. 16, No. 9, of AIAA Journal, 1978
(Revised version of AIAA Paper #77-649)

An integro-differential formulation, previously established for the study of incompressible laminar flow problems, is modified and used in conjunction with the turbulent kinetic energy-energy dissipation rate model for the study of turbulent flow problems. It is demonstrated that the kinematics of the mean turbulent motion is identical to that of the laminar motion. The kinetic aspect of the turbulent flow is significantly more complex. The integro-differential formulation for the turbulent flow retains the important features of the laminar flow and permits the solution field to be confined to the vortical region. Numerical results are presented for several problems involving simple flow boundaries and compared with available experimental data.

INTEGRAL-REPRESENTATION APPROACH FOR TIME-DEPENDENT VISCOUS FLOWS

by J. C. Wu and Y. M. Rizk

To appear in Lecture Notes in Physics, Springer-Verlag, 1978

A novel approach for the numerical simulation of time-dependent viscous flows is described. The distinguishing feature of this approach is the formulation of the entirety of the time-dependent flow problem as integral representations prior to the process of discretization. The approach represents a major extension of the integro-differential approach, also developed by the authors, in which only the kinematic part of the problem is formulated as an integral representation. Entirely new numerical procedures are needed in connection with the present formulation. Several distinct advantages, including the separate handling of the convection of diffusion terms in the transport equation, are offered by the new approach.

NUMERICAL SOLUTION OF UNSTEADY FLOW PROBLEMS USING INTEGRO-DIFFERENTIAL APPROACH

by J. C. Wu, M. M. Wahbah, and A. Sugavanam

To appear in Proceedings of ASME Symposium on Unsteady Fluid Dynamics

A unified numerical approach based on an integro-differential formulation is presented for unsteady incompressible flows past finite solids. The approach

follows the kinetic and kinematic aspects of vorticity dynamics numerically and offers drastic computational advantages through its unique ability to exclude the potential flow region from the solution field. For flows involving two spatial dimensions, a numerical coordinate transformation method is used to map the fluid domain, bounded internally by one or more solid surfaces of any prescribed shape, onto the exterior of a unit circle. Standardized procedures, including a Fourier-series method for the kinematic aspect, are used to solve the problems, which may involve flow turbulence and/or separation, in the circle-plane. Selected numerical results are presented to demonstrate the application of this unified approach under fairly general circumstances.

COMPUTATION OF INTERNAL FLOWS WITH ARBITRARY BOUNDARIES USING THE INTEGRAL REPRESENTATIONS METHOD

by M. M. Wahbah

Georgia Institute of Technology Report, March, 1978

A new method for solving laminar steady flow equations has been developed at Georgia Institute of Technology by Wu and his co-workers. The method employs integral representations of field variables. In this report, the derivation of the matrix equations and the analytical evaluation of the integrals entering the solution procedure are presented. A user-oriented computer program for solving internal flows within arbitrary boundaries is given. The efficiency and the flexibility of the method are demonstrated by presenting several examples of recirculating flows within reasonably complex geometries. A listing of the computer code is given at the end of the report along with some explanations of the code to facilitate the use of the program. The method has been recently extended to turbulent flows in conjunction with a two-equation differential model of turbulence. The results of an illustrative example for turbulent flow is presented here.

A RAREFIED GAS DYNAMIC NUMERICAL METHOD APPLIED TO PROBLEMS IN STATISTICAL TURBULENCE

by R. Srinivasan, D. P. Giddens, L. H. Bangert and J. C. Wu

To appear in Proceedings of the Eleventh International Symposium on Rarefied Gas Dynamics

Lundgren's equation modeling the velocity probability distribution function for turbulence in a parallel flow closely resembles the BGK equation of kinetic theory. The turbulence equation has been solved numerically using a combination of finite differencing and discrete velocities, a method which has seen extensive service in the solution of rarefied gas dynamics problems. Several of the techniques and concepts of rarefield gas dynamics may be transferred to the case of statistical turbulence. This paper discusses some of the similarities and differences between the two model equations and presents results for turbulent Couette flow and channel flow. The turbulence model is shown to give reasonably good results in comparison with experimental data but considerably more research is needed on the question of proper boundary conditions.

PROSPECTS FOR COMPUTATIONAL AERODYNAMICS

by J. C. Wu

NACA CP-2032, pp. 221, February 1978

(Proceedings of NASA Workshop on Future Computer Requirements
for Computational Aerodynamics)

Major difficulties experienced by researchers in the numerical solution of general viscous flow equations are described for high Reynolds number flows. Presently accepted numerical approaches as well as some of the promising alternative approaches, including those developed by the present author, are reviewed. The prospect for the routine numerical prediction of two- and three-dimensional flows involving appreciable regions of separation and high Reynolds number are discussed.

TECHNICAL REPORT

COMPUTATION OF INTERNAL FLOWS WITH ARBITRARY BOUNDARIES USING THE INTEGRAL REPRESENTATIONS METHOD

By

M. M. Wahbah

Prepared for

FLUID MECHANICS BRANCH
U. S. ARMY RESEARCH OFFICE
P. O. BOX 12211
RESEARCH TRIANGLE PARK, N. C. 27709

ARO Grant No. DAAG29–75–G–0147

March 1978

GEORGIA INSTITUTE OF TECHNOLOGY
SCHOOL OF AEROSPACE ENGINEERING
ATLANTA, GEORGIA 30332

Computation of Internal Flows with Arbitrary Boundaries Using the Integral Representations Method

by

M. M. Wahbah
School of Aerospace Engineering
Georgia Institute of Technology

A technical report for research conducted under
ARO Grant No. DAAG29-75-G-0147

March 1978

Unclassified

SECURITY CLASSIFICATION OF THIS PAGE *(When Data Entered)*

REPORT DOCUMENTATION PAGE		READ INSTRUCTIONS BEFORE COMPLETING FORM
1. REPORT NUMBER	2. GOVT ACCESSION NO.	3. RECIPIENT'S CATALOG NUMBER
4. TITLE *(and Subtitle)* COMPUTATION OF INTERNAL FLOWS WITH ARBITRARY BOUNDARIES USING INTEGRAL REPRESENTATIONS METHOD		5. TYPE OF REPORT & PERIOD COVERED Technical 6. PERFORMING ORG. REPORT NUMBER
7. AUTHOR(s) M. M. Wahbah		8. CONTRACT OR GRANT NUMBER(s) ARO Grant No. DAAG29-75-G-0147
9. PERFORMING ORGANIZATION NAME AND ADDRESS Georgia Institute of Technology Atlanta, Georgia 30332		10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS
11. CONTROLLING OFFICE NAME AND ADDRESS Fluid Mechanics Branch U.S. Army Research Office P.O. Box 12211 Research Triangle Park, N.C. 27709		12. REPORT DATE March 1978 13. NUMBER OF PAGES
14. MONITORING AGENCY NAME & ADDRESS*(if different from Controlling Office)*		15. SECURITY CLASS. *(of this report)* Unclassified 15a. DECLASSIFICATION/DOWNGRADING SCHEDULE

16. DISTRIBUTION STATEMENT *(of this Report)*

Approved for public release, distribution unlimited

17. DISTRIBUTION STATEMENT *(of the abstract entered in Block 20, if different from Report)*

18. SUPPLEMENTARY NOTES

19. KEY WORDS *(Continue on reverse side if necessary and identify by block number)*

Viscous Flows
Navier-Stokes Equations
Separated Flows
Computational Fluid Dynamics

20. ABSTRACT *(Continue on reverse side if necessary and identify by block number)*

A new method for solving laminar steady flow equations has been developed at Georgia Institute of Technology by Wu and his co-workers. The method employs integral representations of field variables. In this report, the derivation of the matrix equations and the analytical evaluation of the integrals entering the solution procedure are presented. A user-oriented computer program for solving internal flows within arbitrary boundaries is given. The efficiency and the flexibility of the method are demonstrated by presenting several examples of recirculating flows within reasonably complex geometries. A listing of the

DD FORM 1 JAN 73 1473 EDITION OF 1 NOV 65 IS OBSOLETE

SECURITY CLASSIFICATION OF THIS PAGE *(When Data Entered)*

computer code is given at the end of the report along with some explanations of the code to facilitate the use of the program.

The method has been recently extended to turbulent flows in conjunction with a two-equation differential model of turbulence. The results of an illustrative example for turbulent flow is presented here.

ACKNOWLEDGMENTS

The author is thankful to professor J.C. Wu for suggesting the present analysis, and for many useful disscussions regarding this endeavor and other aspects of computational fluid dynamics.

The financial support of this work by the Army Research Office, Grant No. DAAG29-75-G-0147, is gratefully acknowledged.

TABLE OF CONTENTS

SUMMARY

A new method for solving laminar steady flow equations has been developed at Georgia Institute of Technology by Wu and his co-workers. The method employs integral representations of field variables. In this report, the derivation of the matrix equations and the analytical evaluation of the integrals entering the solution procedure are presented. A user-oriented computer program for solving internal flows within arbitrary boundaries is given. The efficiency and the flexibility of the method are demonstrated by presenting several examples of recirculating flows within reasonably complex geometries. A listing of the computer code is given at the end of the report along with some explanations of the code to facilitate the use of the program.

The method has been recently extended to turbulent flows in conjunction with a two-equation differential model of turbulence. The results of an illustrative example for turbulent flow is presented here.

1. Introduction.

During the past few years, new techniques for solving viscous flow problems have been developed, calibrated, and applied to several important problems at the Georgia Institute of Technology. The distinguishing feature of these techniques is the replacement of the more familiar governing differential equations by equivalent integral representations.

Wu and his co-workers, references [1 to 5], initiated these research efforts, by first developing the Integro-Differential method for the treatment of time-dependent viscous flows involving appreciable flow separations [1]. In this method, only the kinematic part of the governing equations was cast into an integral representation, and steady solutions, if desired, were obtained as asymptotic development of the time dependent problem. The method proved to possess several advantages: (1) The solution field can be confined to the viscous region of the flow, (2) the confined solution field can be segmented and each segment treated independently, (3) numerical boundary conditions that presented difficulties in previous methods can be treated in a precise manner. As a result, the method proved to offer remarkable solution speed and accuracy.

These efforts were recently extended to directly treat steady flows and a new method, called the Integral-Representations method, was developed (2). The new method retains the above mentioned desirable features. In this method, the complete set of the governing equations are recast into equivalent integral representations of the field variables. The use of the velocity and the vorticity vectors as dependent variables allows the

partition of the problem into two recognizable parts: kinematics and kinetics. The reader is referred to references 1 to 5 for the derivation, properties, and the unique features of this method.

This report is concerned with the derivation of the matrix equations and the analytical evaluation of the integrals associated with the use of a finite element method for iteratively solving the integral representations in two-dimensional incompressible laminar flows. The term finite element method is used here in its current broad context to indicate a numerical procedure which involves the division of the solution field into elements (subregions) each associated with a finite number of data nodes. A user-oriented computer program for solving internal flows within arbitrary boundaries is presented. The program requires only the input of the nodes of the finite element mapping and the velocity values at the boundary nodes. No additional efforts, such as special techniques for curved boundaries, are required. In the resulting matrix equations, all the coefficient matrices are purely geometric. The elements of the matrices depend only on the relative locations between the nodes. Consequently, these matrices can be computed once, stored, and used in the various iterations for any given problem. Furthermore, for problems with the same boundary configuration but different flow conditions at the boundary or different flow Reynolds numbers, the matrices entering the solution procedure are identical. These features are utilized to drastically reduce the required computation. For example, in obtaining a solution for a problem at a high Reynolds number, one first obtains a solution for the problem at a lower Reynolds number. This lower Reynolds number solution is then utilized as the initial solution to start the iteration procedure for the high Reynolds number case. This process of escalating the Reynolds number can be utilized

to obtain a set of solutions for various Reynolds numbers. The total number of iterations required to solve both the high and lower Reynolds number problems is generally much smaller than the number of iterations required to solve the high Reynolds number case alone starting with an arbitrarily assumed initial solution. Since the matrices need not be recalculated for the two problems, the computing time required by the Reynolds number escalating approach is generally much smaller than that required for solving only the high Reynolds number problem.

The method described in this report permits the use of polygonal elements with any finite number of sides associated with element interpolation-functions of any order. The use of first order triangular elements is emphasized in the report because of its relative simplicity. The user oriented computer program, presented as Appendix C of this report, is based on the use of first order triangular elements. Analytical expressions for higher order (as well as first order) polygonal elements are derived and presented in Appendix A. Exploratory studies show that the use of second order elements further improves computational efficiency. For the type of problems considered thus far at Georgia Tech, the amount of computation required is small and it is not essential to go beyond the first order elements. It is worthy of note, however, that for future applications requiring larger amounts of computation, it may be desirable to incorporate the option of using higher order elements into the computer program. A subroutine for evaluating certain integrals related to the use of polygonal elements (including triangular elements) of any finite order (including the first order) is presented as a part of Appendix A for this purpose.

It is well known that the majority of viscous flow problems occuring in nature and in applications involve high Reynolds number and flow turbulence. The lack of efficient and accurate solution algorithm for flow equations has been a major obstacle in the development of suitable turbulence models for separated flows. One of the objectives of the present effort is to provide such a solution algorithm for two-dimensional steady separated flows so that the extensive numerical experimentation and calibration required for the development of suitable turbulence models may proceed for such flow. At Georgia Tech, such experimentation and calibration has been initiated using the approach described in the report in conjunction with a two-equation differential model of turbulence (6). The results of an illustrative problem for turbulent flow is presented.

2. Integral Representations Method

The velocity vector $\vec{v}_o$ and the vorticity vector $\vec{\omega}_o$ at any point $\vec{r}_o$, in a region R surrounded by a surface B, are given by the following integral representations [2]

$$\vec{v}_o = \frac{-1}{A} \int_R \frac{\vec{\omega} \wedge (\vec{r}-\vec{r}_o)}{|\vec{r}-\vec{r}_o|^d} \, dR + \frac{1}{A} \int_B \frac{[(\vec{v}.\vec{n}) - (\vec{v}\wedge\vec{n})\wedge] \, (\vec{r}-\vec{r}_o)}{|\vec{r}-\vec{r}_o|^d} \, dB \qquad (1)$$

$$\vec{\omega}_o = \frac{-1}{A\nu} \int_R \frac{(\vec{v}\wedge\vec{\omega}) \wedge (\vec{r}-\vec{r}_o)}{|\vec{r}-\vec{r}_o|^d} \, dR + \frac{1}{A} \int_B \frac{[(\quad h \quad)\vec{n}\wedge + (\vec{\omega}.\vec{n}) - (\vec{\omega}\wedge\vec{n})\wedge] \, (\vec{r}-\vec{r}_o)}{|\vec{r}-\vec{r}_o|^d} \, dB \qquad (2)$$

where $\vec{r}$ is a position vector, ν is the fluid kinematic viscosity, $\vec{n}$ is the outward normal, $A = 4\pi$ and $d = 3$ for three-dimensional problems, $A = 2\pi$ and $d = 2$ for two-dimensional problems, and h is the

total pressure divided by ν and is defined by

$$h = \left(\frac{p}{\rho} + \frac{V^2}{2}\right) / \nu \quad ; \quad V = \left| \vec{v} \right|$$

where p is the pressure and ρ is the fluid density.

Equation (1) is completely equivalent to the continuity equation $\vec{\nabla} \cdot \vec{v} = 0$, along with the vorticity definition $\vec{\omega} = \vec{\nabla} \times \vec{v}$. The velocity boundary condition is incorporated in this equation and is given by the boundary integral. Equation (2) replaces the more familiar Navier-Stokes momentum equations for an incompressible flow.

Equation (1) permits the explicit, point by point, computation of $\vec{v}_o$ in R using the values of $\vec{\omega}$ in R and the velocity boundary condition on B. Equation (2) allows the computation of $\vec{\omega}$ in R using values of $\vec{v}$ and $\vec{\omega}$ in R and the vorticity boundary condition on B. Due to the existing non-linearity in equation (2), the two equations are solved iteratively, as will be explained in details in later sections, starting with assumed values of $\vec{\omega}$ in R.

For two-dimensional flows, equations (1) and (2) can be written as

$$u_o = \frac{1}{2\pi} \iint_R \omega \frac{Y}{X^2 + Y^2} dxdy + \frac{1}{2\pi} \int_B \left(v_n \frac{X}{X^2 + Y^2} - v_t \frac{Y}{X^2 + Y^2} \right) ds \qquad (3)$$

$$v_o = \frac{-1}{2\pi} \iint_R \omega \frac{X}{X^2 + Y^2} dxdy + \frac{1}{2\pi} \int_B \left(v_n \frac{Y}{X^2 + Y^2} + v_t \frac{X}{X^2 + Y^2} \right) ds \qquad (4)$$

$$\omega_o = \frac{-1}{2\pi\nu} \iint_R \left(\omega u \frac{X}{X^2+Y^2} + \omega v \frac{Y}{X^2+Y^2} \right) dxdy + \frac{1}{2\pi} \int_B h \frac{(\vec{r}-\vec{r}_o).\vec{t}}{X^2+Y^2} ds + \frac{1}{2\pi} \int_B \omega \frac{(\vec{r}-\vec{r}_o).\vec{n}}{X^2+Y^2} ds \qquad (5)$$

where u and v are the velocity components in the x and y directions, $\vec{t}$ is the unit tangent to the boundary B, s is the distance along B, Figure (1), the subscripts n and t indicate respectively the normal and the tangential components of a vector quantity, and

$$X = x - x_o$$

$$Y = y - y_o \qquad (6)$$

3. Evaluation of the Integrals

The integrals in equations (3-5) are best performed by using the finite element technique. The region R is mapped into elements, Figure (2), the number, the shape, and the distribution of these elements depend upon the particular problem to be solved and the degree of accuracy desired. The current procedure allows for the arbitrary mapping of the region into polygonal elements with any number of sides, associated with element interpolation-functions of any order. The present report emphasizes the use of first order triangular elements. Extensions to higher order elements, which can be easily incorporated in the program, are pointed out. The computer code at the end of this report is for the first order case.

3.1 Evaluation of the Area Integrals

The area integrals in equation (3-5) can be expressed as the sum of contributions from individual elements, designated by (e). Consider for example the area integral in equation 3 which can be written as

$$UR_o = \iint_R \omega \frac{Y}{X^2 + Y^2} \, dxdy$$

$$= \sum_e \iint_e \omega^e \frac{Y}{X^2 + Y^2} \, dxdy \qquad (7)$$

For simplicity, we consider triangular elements with three nodes located at the triangle vertices, Fig. (3). The contribution to UR_o from one element, denoted by UR_o^e, is given by

$$UR_o^e = \iint_{\bar{e}} \omega^e \frac{Y}{X^2+Y^2} \, dxdy \tag{8}$$

The element interpolation function is then introduced and ω^e is expressed as

$$\omega^e = a^e + b^e x + c^e y \tag{9}$$

Using equation (6) we obtain

$$\omega^e = A^e + B^e X + C^e Y \tag{10}$$

where

$$A^e = a^e + b^e x_o + c^e y_o \tag{11}$$

$$B^e = b^e$$

$$C^e = c^e$$

Substituting equation (10) in equation (8), we get

$$UR_o^e = A^e I_{o,1}^e + B^e I_{1,1}^e + C^e I_{o,2}^e \tag{12}$$

where

$$I_{n,m}^e = \iint_{\bar{e}} \frac{X^n Y^m}{X^2+Y^2} \, dXdY \tag{13}$$

An exact analytical expression was derived for $I_{n,m}^e$ [7]. The expression is valid for any combination of n and m, and the element (e) may be any polygon with any number of sides. The result is expressed in terms of the coordinates of the polygon vertices and the coordinates of the point (x_o, y_o). The analytical derivation is given in Appendix A. A computer code has been developed to efficiently evaluate these functions (Subroutine GF of the computer code).

From equations (7), (8), and (12) we obtain

$$UR_o = \sum_e \left[I^e_{o,1} \; a^e + (x_o \; I^e_{o,1} + I^e_{o,1}) \; b^e + (y_o \; I^e_{o,1} + I^e_{o,2}) \; c^e \right] \qquad (14)$$

The constants a^e, b^e and c^e can be expressed in terms of ω_i ($i = 1, 2, 3$), the values of ω at the nodes i of the element (e), Fig. (4), by satisfying equation (9) at the element nodes. The result is

$$\begin{aligned} a^e &= \sum a^e_i \; \omega_i \\ b^e &= \sum b^e_i \; \omega_i \\ c^e &= \sum c^e_i \; \omega_i \end{aligned} \qquad i = 1,2,3 \qquad (15)$$

where

$$\begin{aligned} a^e_i &= (x_j y_k - x_k y_j) / D \\ b^e_i &= (y_j - y_k) / D \\ c^e_i &= (x_k - y_j) / D \end{aligned} \qquad (16)$$

Furthermore

$$D = (x_2 y_3 - x_3 y_2) + (x_3 y_1 - x_1 y_3) + (x_1 y_2 - x_2 y_1) \qquad (17)$$

The subscripts i, j and k, Eq. (16), should be taken in the cyclic order (1-2-3-1).

Substituting equation (15) into equation (14), gives

$$UR_o = \sum_e \left\{ \sum_{i=1}^{3} \left[I^e_{o,1} \; a^e_i + (x_o \; I^e_{o,1} + I^e_{1,1}) \; b^e_i + (y_o \; I^e_{o,1} + I^e_{o,2}) \; c^e_i \right] \omega_i \right\} \qquad (18)$$

It should be noted that in equations (15) through (18) the indices and subscripts used for x, y and ω are local indices pertinent to the particular element under consideration, and take the values 1, 2 and 3. Hence, each node may have different local element-index when considering the different elements meeting at this particular node.

In equation (18), summing over all elements and adding up the coefficients of ω_n for each node n in the complete grid, contributed by all elements meeting at this node, we obtain

$$UR_o = \sum_{n=1}^{N} D_{o,n} \omega_n \qquad (19)$$

and

$$D_{o,n} = \sum_{e} \left[I^e_{o,1} a^e_i + (x_o I^e_{o,1} + I^e_{1,1}) b^e_i + (y_o I^e_{o,1} + I^e_{o,2}) c^e_i \right] \qquad (20)$$

where i is the local index of the node n in the element e, and N is the total number of nodes in the grid system. The summation in equation (20) is performed over all elements meeting at the node n. In matrix notation, equation (19) becomes

$$UR_o = \left[D_{o,n} \right] \left\{ \omega_n \right\} \qquad (21)$$

where $\left[D_{o,n} \right]$ is a (1XN) row matrix and $\left\{ \omega_n \right\}$ is a (N) column vector.

Similarly, the area integral in the expression for v, Eqn. (4), can be written as

$$VR_o = \left[E_{o,n}\right] \quad \{\omega_n\} \tag{22}$$

where the elements of $\left[E_{o\,n}\right]$ are given by

$$E_{o,n} = \sum_e \left[I^e_{1,o}\ a^e_i + (x_o\ I^e_{1,o} + I^e_{2,o})\ b^e_i + (y_o\ I^e_{1,o} + I^e_{1,1})\ c^e_i \right] \tag{23}$$

In equation (5), the area integral can be evaluated by introducing the element interpolation-functions, similar to equations (9) and (10), for the field variables $(\omega u)^e$ and $(\omega v)^e$. The choice of these combinations lead to coefficient matrices that are not only independent of the field variables, but also equivalent to those in equations (21) and (22). A direct comparison between the area integrals in equations (3) through (5), shows that the area integral in equation (5) can be written as

$$\omega R_o = \left[E_{o,n}\right] \quad \left\{(\omega u)_n\right\} + \left[D_{o,n}\right]\left\{(\omega v)_n\right\} \tag{24}$$

where $(\omega u)_n$ and $(\omega v)_n$ are the values of ωu and ωv at the node n, respectively, and $D_{o,n}$ and $E_{o,n}$ are given by equations (20) and (23).

The introduction of higher order elements, e.g. second order triangular elements with six nodes as shown in Fig. (4), or a general polygon with arbitrary number of sides and nodes, represents no difficulty and can be carried out using the same procedure described above for the first order triangular elements. For example, if ω^e is represented by

$$\omega^e = \sum_{n,m} K^e_{n,m}\ X^n\ Y^m \tag{25}$$

then, the area integrals in equations (3) and (4) take the form

$$UR_o = \sum_e \left(\sum_{n,m} K^e_{n,m} \; I^e_{n,m+1} \right) \tag{26}$$

$$VR_o = \sum_e \left(\sum_{n,m} K^e_{n,m} \; I^e_{n+1,m} \right) \tag{27}$$

The constants $K^e_{n,m}$ are evaluated by satisfying equation (25) at the element nodes. The equivalent matrix forms, similar to equations (21) and (22), are obtained in exactly the same manner as explained earlier.

For a discussion on the completeness and efficiency of various finite-element interpolation functions, associated with different finite elements, the reader is referred to reference [8] .

3.2 Evaluation of the Boundary Integrals

The method is explained for the boundary integral in uo expression, equation (3), and is then extended to other integral.

3,2a Evaluation of the Boundary Integral in u_o Expression

The integral is given by

$$UB_o = \int_B \left(v_n \frac{X}{X^2 + Y^2} - v_t \frac{Y}{X^2 + Y^2} \right) ds \tag{28}$$

The above line-integral can be evaluated in a closed form for several particular cases. In the following analysis, a more general procedure is described for the numerical evaluation of this type of integral. The procedure, in addition to being general for any boundary and any velocity distribution on the boundary, results in geometric coefficient matrices that are equivalent to others, to be derived in later sections, in the boundary contribution to ω_o, equation (5).

The boundary B is divided into straight line segments, Fig. (5). The method is first explained for linear variations, of field variables, along those line segments. Extensions to higher order variations are straightforward. For linear variations, each line segment is associated with two boundary nodes at both ends of the segment. The number of line segments is N_b, where N_b is the number of boundary nodes. These boundary nodes are taken to be the nodes of the finite element mapping lying on the boundary.

The boundary integral can be expressed as the sum of all contributions from the individual line segments, designated by the subscript k. Then

$$UB_o = \sum_{k=1}^{N_b} \int_o^{l_k} \frac{(v_n X - v_t Y)}{X^2 + Y^2} \, ds \qquad (29)$$

where l_k is the length of the segment k, and v_t and v_n are the tangential and the normal velocity components along the segment k, respectively.

The contribution from each line segment is best evaluated by using the coordinates (x', y') depicted in Figure (5). The origin of these coordinates lies at end (k) of the line L_k, and the x' axis is directed along the line segment itself, in the counterclockwise direction of the curve B. Further, it is more convenient to use u and v instead of v_t and v_n.

The unit normal to the curve B is given by

$$\vec{n} = n_x \vec{i} + n_y \vec{j} \qquad (30)$$

where the subscripts x and y denote components along the x and y axes respectively. It is easy to show that

$$\begin{aligned} \vec{t} &= t_x \vec{i} + t_y \vec{j} \\ &= -n_y \vec{i} + n_x \vec{j} \end{aligned} \qquad (31)$$

Hence

$$v_n = u\, n_x + v\, n_y \qquad (32)$$
$$v_t = -u\, n_y + v\, n_x$$

Using equation (32), the numerator in equation (29) can be written as

$$\begin{aligned} v_n X - v_t Y &= u(n_x X + n_y Y) + (n_y X - n_x Y) \\ &= u(\vec{n} \cdot \vec{R}) - v(\vec{t} \cdot \vec{R}) \end{aligned} \qquad (33)$$

where

$$\vec{R} = X\,\vec{i} + Y\,\vec{j} \qquad (34)$$

The quantities $(\vec{n}.\vec{R})$ and $(\vec{t}.\vec{R})$ are the components of the vector $\vec{R}$ in the directions $\vec{n}$ and $\vec{t}$, respectively, and can be written as

$$\begin{aligned} (\vec{n} \cdot \vec{R}) &= y' - y_o' \\ (\vec{t} \cdot \vec{R}) &= x' - x_o' \end{aligned} \qquad (35)$$

where x_o' and y_o' are the coordinates of the point (o) with respect to the axes (x',y') at the point (k). Also,

$$\begin{aligned} R^2 &= X^2 + Y^2 \\ &= (x' - x_o')^2 + (y' - y_o')^2 \end{aligned} \qquad (36)$$

Using equations (33), (35) and (36), the boundary integral UB, equation (29), can be written in the form

$$UB_o = \sum_{k=1}^{N_b} \left[-\int_o^{1^k} u\, \frac{y_o'}{R^2}\, dx' - \int_o^{1^k} v\, \frac{(x' - x_o')}{R^2}\, dx' \right] \qquad (37)$$

The fact that $y' = o$, on each segment k, has been used in writing equation

(37). For the purposes of analysis, we write

$$UB_o = I_u + I_v \tag{38}$$

where

$$I_u = \sum_{k=1}^{N_b} I_u^k \tag{39}$$

$$I_v = \sum_{k=1}^{N_b} I_v^k \tag{40}$$

Furthermore,

$$I_u^k = - \int_o^{l_k} u \frac{y_o'}{R^2} \, dx' \tag{41}$$

$$I_v^k = - \int_o^{l_k} v \frac{(x' - x_o')}{R^2} \, dx' \tag{42}$$

In the following analysis, the quantities I_u^k and I_v^k are evaluated in terms of u and v, respectively, at the end points k and $(k+1)$ of the line segment. The method is demonstrated for I_u^k. For linear variations, along the line segment, u can be expressed as

$$u = A (x' - x_o') + B \tag{43.a}$$

The constants A and B can be expressed in terms of u_k and u_{k+1} the values of u at the nodes k and $k+1$ by satisfying the above expression at the segment nodes, giving

$$A = (u_{k+1} - u_k) / l_k$$
$$B = u_k + A x_o' \tag{43-b}$$

Substituting equation (43-a) into equation (41), and carrying out the resulting integrals, give

$$I_u^k = - (A y_o') f_1^k - B f_o^k \tag{44}$$

where

$$f_o^k = \tan^{-1} \frac{(l_k - x'_o)}{y'_o} + \tan^{-1} \frac{x'_o}{y'_o}$$

$$f_1^k = \tfrac{1}{2} \ln \left[\frac{(l_k - x'_o)^2 + y'^2_o}{x'^2_o + y'^2_o} \right] \qquad (45)$$

The function f_o^k can be expressed in the following form

$$f_o^k = \tan^{-1} \left[\frac{l_k \, y'_o}{y'^2_o - (l_k - x'_o) \, x'_o} \right] \qquad (46)$$

with the principal branch of the inverse tangent function defined as $(-\pi < \tan^{-1} \frac{x}{y} < \pi)$, taking into consideration the individual signs of the numerator x, and the denominator y. The inverse tangent, defined in this manner, is usually supplied on many computers as ATAN2 (x,y). The reason for deriving the second form of f_o^k, is to avoid the excessive use of the time consuming trigonometric functions.

Substitution of equations (43-b) into equation (44), gives

$$I_u^k = F_1^k \, u_k - F_2^k \, u_{k+1} \qquad (47)$$

where

$$F_1^k = \frac{y'_o}{l_k} f_1^k + \left(\frac{x'_o}{l_k} - 1 \right) f_o^k \qquad (48)$$

$$F_2^k = \frac{y'_o}{l_k} f_1^k + \frac{x'_o}{l_k} f_o^k$$

Equations (39) and (46) yield

$$I_u = \sum_{k=1}^{N_b} (F_1^k \, u_k - F_2^k \, u_{k+1}) \qquad (49)$$

In equation (48), summing over all line segments and adding up the coefficient of u_n $(n = 1, N_b)$ for each boundary node n, contributed by the two line segments $(n-1, n)$ meeting at this node, give

$$I_u = \sum_{n=1}^{N_b} G_{o,n} \, u_n \tag{50}$$

with

$$G_{o,n} = F_1^{\,n} - F_2^{\,n-1} \tag{51}$$

where the line segment (n) is assumed to succeed the line segment (n-1) when going around B in the counterclockwise direction.

It should be noted that the quantities f_o^k and f_1^k, equation (45), possess singular behavior as the point (x_o, y_o) coincides with one of the end points of the line segment k. The total contribution from the two line segments meeting at this point, however, is a definite quantity given by the limiting value of $G_{o,n}$, equation (51), as the point o coincides with point (n), Appendix B.

The contribution of the u velocity-component to UB_o, equation (37), can be written in the following matrix form

$$Iu = \left[G_{o,n}\right] \quad \{u_{bn}\} \tag{52}$$

where $\left[G_{o,n}\right]$ is a $(1 \times N_b)$ row matrix and $\{u_{bn}\}$ is a (N_b) column matrix, representing the values of u at the N_b boundary points. The subscript b has been added to u_n to distinguish it from $\{u_n\}$, $(n = 1, N)$, to be introduced later.

The above developments can be repeated, almost identically, to compute the contribution of the v velocity-component to UB_o. The result can be represented in a matrix form, similar to equations (49) and (52), as

follows

$$I_v = \sum_{k=1}^{N_b} (F_3^k v_k - F_4^k v_{k+1}) \qquad (53\text{-a})$$

and $$I_v = \left[H_{o,n}\right]\{v_{bn}\} \qquad (53\text{-b})$$

where the summation in (53-a) is over the line segments forming the boundary, $\{v_{bn}\}$ is a (N_b) column vector, representing the values of v at the N_b boundary points, and $\left[H_{o,n}\right]$ is a $(1 \times N_b)$ row matrix whose elements are given by $$H_{o,n} = F_3^n - F_4^{n-1} \qquad (54)$$

With

$$F_3^k = 1 - \frac{y'_o}{l_k} f_o^k + \left(\frac{x'_o}{l_k} - 1\right) f_1^k$$

$$F_4^k = 1 - \frac{y'_o}{l_k} f_o^k + \frac{x'_o}{l_k} f_1^k \qquad (55)$$

The functions f_o^k and f_1^k in the above two equations are those defined previously by equation (45), and possess singular behavior as the point o coincides with the boundary point (n). The limiting value of $H_{n,n}$ is obtained in Appendix B

Finally, substituting equations (52) and (53-b) in equation (38), gives

$$UB_o = \left[G_{o,n}\right] \{u_{bn}\} + \left[H_{o,n}\right] \{v_{bn}\} \qquad (56)$$

The same procedure can be used for higher order interpolation-functions, along line segments associated with several nodes. Analogizing to equation (39), we may have

$$u = \sum_{m=o}^{M} A_m (x' - x'_o)^m$$

$$v = \sum_{m=o}^{M} B_m (x' - x'_o)^m \qquad (57)$$

The number of nodes on the line segment should be equal to (M+1). Different segments may have different number of nodes corresponding to different order of interpolation along them. Substituting equation (57) into equations (41) and (42), give

$$I_u^k = \sum_{m=o}^{M} (A_m y_o'^m) f_m$$
$$I_v^k = \sum_{m=o}^{M} (B_m y_o'^{(m-1)}) f_{m+1} \qquad (58)$$

where f_o and f_1 are given by equation (45). For $m \geq 2$, the functions f_m are given by the following recurrence relation

$$f_m = \frac{1}{m-1}\left[\frac{(l_k - x'_o)^{m-1}}{y'_o} - \frac{(-x'_o)^{m-1}}{y'_o}\right] - f_{m-2} \quad (m \geq 2) \qquad (59)$$

The final matrix form of UB_o, corresponding to equation (56), can be obtained by following the same steps explained earlier for linear variations along the line segments.

3.2b Evaluation of the Boundary Integral in v_o expression

The integral is given by

$$VB_o = \int_B \left(v_n \frac{Y}{X^2+Y^2} + v_t \frac{X}{X^2+Y^2}\right) ds \qquad (60)$$

Using the same technique, section 3.2.1, the above integral can be re-expressed in terms of u_b and v_b. The same coordinates (x',y'), depicted in Figure (5), are employed for the present analysis. The result, corresponding to equation (37), is

$$VB_o = \sum_{k=1}^{N_b} \left[\int_o^{l_k} u \frac{(x'-x'_o)}{R^2} dx' - \int_o^{l_k} v \frac{y'_o}{R^2} dx'\right] \qquad (61)$$

Comparing equation (61) with equation (37), it can be easily seen that the influence-coefficients matrix associated with u in equation (61) is the same as that associated with v in equation (37), but with a different sign, and the influence-coefficients matrix associated with v in equation(61)is the same as that associated with u in equation (37). Consequently the matrix representation of VB_o is given by

$$VB_o = - \left[H_{o,n}\right] \{u_{bn}\} + \left[G_{o,n}\right] \{v_{bn}\} \qquad (62)$$

The matrices and vectors appearing in equation (62), are those defined earlier in connection with equation (56). The equivalence of some influence-coefficients matrices, which occurs in several places in the present technique, is a very desirable feature, which helps to reduce the required computer time and storage.

3.3 Evaluation of the Boundary Integral in ω_o expression

The integral is given by

$$WB_o = \int_B h \frac{(\vec{r} - \vec{r}_o).\vec{t}}{X^2 + Y^2} ds + \int_B \omega \frac{(\vec{r} - \vec{r}_o).\vec{n}}{X^2 + Y^2} ds \qquad (63)$$

The integral is evaluated as the sum of integrals over the individual line segments, using the segment local-coordinates (x',y'), giving

$$WB_o = \sum_{k=1}^{N_b} \left[\int_o^{l_k} h \frac{(x' - x_o')}{R^2} dx' - \int_o^{l_k} \omega \frac{y_o'}{R^2} ds \right] \qquad (64)$$

Comparing equations (64) with equation (61), it can be easily seen that the influence coefficient matrices associated with h and ω, in equation (64), are identical to those associated with u and v, in equation (61), respectively. Hence, the matrix representation of WB_o can be written in the form,

$$WB_o = -\left[H_{o,n}\right]\left\{h_{bn}\right\} + \left[G_{o,n}\right]\left\{\omega_{bn}\right\} \tag{65}$$

where $\left\{h_{bn}\right\}$ is a (N_b) column vector representing the values of h at the N_b boundary points and ω_{bn} is a (N_b) column vector whose elements are the values of ω at those points.

4. The Matrix Form of the Integral Representations

In the present section, the different integrals appearing in the original integral representations are grouped together and the elements of the corresponding matrices are collected below for the convenience of reference.

Substituting equations (21), (22), (24), (56), (62), and (65) into the original integral representations (3) through (5), gives

$$2\pi u_o = \left[D_{o,n}\right]\left\{\dot{\omega}_n\right\} + UB_o \tag{66}$$

$$2\pi v_o = -\left[E_{o,n}\right]\left\{\omega_n\right\} + VB_o \tag{67}$$

$$2\pi \omega_o = \frac{-1}{\nu}\left[E_{o,n}\right]\left\{(\omega u)_n\right\} - \frac{1}{\nu}\left[D_{o,n}\right]\left\{(\omega v)_n\right\} - \left[H_{o,n}\right]\left\{h_{bn}\right\} + \left[G_{o,n}\right]\left\{\omega_{bn}\right\} \tag{68}$$

with

$$UB_o = \left[G_{o,n}\right]\left\{u_{bn}\right\} + \left[H_{o,n}\right]\left\{v_{bn}\right\} \tag{69}$$

$$VB_o = -\left[H_{o,n}\right]\left\{u_{bn}\right\} + \left[G_{o,n}\right]\left\{v_{bn}\right\} \tag{70}$$

The above set of equations can be written for any point (o) in the domain R, giving

$$2\pi \{u\} = [D] \{\omega\} + \{UB\} \quad (71)$$

$$2\pi \{v\} = -[E]\{\omega\} + \{VB\} \quad (72)$$

$$2\pi \{\omega\} = -\frac{1}{\nu} [E] \{\omega u\} - \frac{1}{\nu} [D] \{\omega v\} - [H] \{h_b\} + [G] \{\omega_b\} \quad (73)$$

With

$$\{UB\} = [G] \{u_b\} + [H]\{v_b\} \quad (74)$$

$$\{VB\} = -[H] \{u_b\} + [G]\{v_b\} \quad (75)$$

where $\{u\}$, $\{v\}$, $\{\omega\}$, $\{\omega u\}$ and $\{\omega v\}$ are (N) column vectors representing the values of u, v, ω, ωu and ωv, respectively, at the N grid nodes; $\{u_b\}$, $\{v_b\}$, $\{\omega_b\}$ are (N_b) column vectors representing the corresponding variables at the N_b boundary nodes; [D] and [E] are (NXN) geometric-coefficients matrices; and [G] and [H] are (NXN_b) geometric-coefficients matrices. The m^{th} rows of [D], [E], [G] and [H] are the previously developed row matrices $[D_{o,n}]$, $[E_{o,n}]$, $[H_{o,n}]$ respectively, with the point o being the point (m). The elements of these matrices are given by

$$D_{m,n} = \Sigma_e \left[I^e_{o,1} a_i + (x_m I^e_{o,1} + I^e_{1,1}) b^e_i + (y_m I^e_{o,1} + I^e_{o,2}) c^e_i \right] \quad (76)$$

$$E_{m,n} = \Sigma_e \left[I^e_{1,o} a^e_i + (x_m I^e_{1,o} + I^e_{2,o}) b^e_i + (y_m I^e_{1,o} + I^e_{1,1}) c^e_i \right] \quad (77)$$

with

$$I^{e}_{m,n} = \iint_{e} \frac{X^{m}\, Y^{n}}{X^{2}+Y^{2}}\, dXdY \tag{78}$$

$$a^{e}_{i} = (x_j\, y_k - x_k\, y_j)\ /D \tag{79}$$

$$b^{e}_{i} = (y_j - y_k)\ /D \tag{80}$$

$$c^{e}_{i} = (x_k - x_j)\ /D \tag{81}$$

where

$$D = (x_2y_3 - x_3y_2) + (x_3y_1 - x_1y_3) + (x_1y_2 - x_2y_1) \tag{82}$$

Furthermore

$$G_{m,n} = \left[\frac{y'_m}{l_n}\, f_1^n + \left(\frac{x'_m}{l_n} - 1\right) f_o^n \right] - \left[\frac{y''_m}{l_{n-1}}\, f_1^{n-1} + \frac{x''_m}{l_{n-1}}\, f_o^{n-1} \right] \tag{83}$$

$$H_{m,n} = \left[1 - \frac{y'_m}{l_n}\, f_o^n + \left(\frac{x'_m}{l_n} - 1\right) f_1^n \right] - \left[1 - \frac{y''_m}{l_{n-1}}\, f_o^{n-1} + \frac{x''_m}{l_{n-1}}\, f_1^{n-1} \right] \tag{84}$$

where (x", y") are the coordinates of the point m with respect to the axes (x",y"), pertinent to the line segment (n-1), Figure 5, and

$$f_o^n = \tan^{-1} \frac{l_n - x'_m}{y'_m} + \tan^{-1} \frac{x'_m}{y'_m} \tag{85}$$

$$f_1^n = \frac{1}{2} \ln \left[\frac{(l_n - x'_m)^2 + y'^2_m}{x'^2_m + y'^2_m} \right] \tag{86}$$

It should be noted that the four geometric coefficients matrices depend only on the relative locations between the grid nodes and are independent of the field variables (u, v, and ω) and of the boundary conditions. This feature has several advantages resulting in a drastic reduction in the required computer time and storage. The matrices can be computed, stored and used in the solution of different flow problems, within the same domain R, for different boundary conditions and different Reynolds numbers. Besides, they need not be computed repeatedly from iteration to iteration during the iterative procedure for solving the governing equations. For a spcified velocity boundary condition (u_b, v_b), the vectors

$\{UB\}$ and $\{VB\}$ become constant vectors and are computed only once, from equations (74) and (75), during the solution procedure.

5. Solution Procedure

In this section, a general computation procedure for solving internal flow problems within arbitrary boundaries, is presented. The velocities u and v at the boundary nodes are assumed to be known. Extensions to other boundary conditions, and to external flows, can be easily incorporated.

Equations (71) and (72) permit the explicit, point by point, computation of u and v in R using the values of ω in R and the velocity boundary condition (u_b, v_b) on B. Equation (73) allows the determination of ω in R using the values of u and v in R and the boundary vorticity ω_b. Due to the existing non-linearity in equation (73), the three equations are solved iteratively.

Starting with known values of $\{\omega\}^n$, the superscript n being the iteration counter, at the $(N-N_b)$ interior nodes in R, the following steps constitute one iteration loop

(1). Computation of the boundary vorticity $\{\omega_b\}^{n+1}$

The proper velocity boundary condition for the interior flow problem is either the tangential velocity component on B or the normal velocity component on B plus the circulation around B [5]. In the present analysis the tangential velocity is used. The requirement that the tangential velocity obtained form equations (71) and (72) be equal to the prescribed tangential velocity at the N_b boundary nodes gives a set of N_b algebraic equations in which $\{\omega_b\}$ is the only unknown. The tangential velocity at the N_b boundary points is given by

$$\{v_t\} = \{t_x u_b + t_y v_b)\} \tag{87}$$

where u_b, v_b are the prescribed u and v on the boundary, respectively. Using equations (71) and (72), we get

$$2\pi \{v_t\} = [S]\{\omega\} + \{T\} \tag{88}$$

where

$$\left.\begin{aligned} S_{m,n} &= (t_{xm} D_{m,n} - t_{ym} E_{m,n}) \qquad 1 \le m \le N_b \qquad (89)\\ T_m &= (t_{xm} UB_m + t_{ym} VB_m) \qquad 1 \le m \le N_b \qquad (90)\\ &\qquad\qquad 1 \le n \le N \end{aligned}\right\} ;$$

Assuming the grid nodes, whose total number is N, to be indexed such that the first N_b points are the N_b boundary nodes, the first term on the righthand side can be split in the following manner

$$[S]\{\omega\} = [SB]\{\omega_b\} + [SI]\{\omega_I\} \tag{91}$$

where

$$SB_{m,n} = S_{m,n} \qquad 1 \le n \le N_b \tag{92}$$

$$SI_{m,n} = S_{m,n+N_b} \qquad 1 < n \le (N-N_b) \tag{93}$$

and

$$\omega_{In} = \omega_{n+N_b} \qquad 1 < n \le (N-N_b) \tag{94}$$

Substituting equations (87) and (91) into equations (88), and rearranging, gives

$$[SB]\{\omega_b\} = 2\pi\{(t_x u_b + t_y v_b)\} - \{T\} - [SI]\{\omega_I\} \tag{95}$$

The above equation can be solved for $\{\omega_b\}^{n+1}$ using $\{\omega_I\}^n$.

The coefficient matrix [SB] does not change between iterations, therefore, the Gauss elmination procedure needs to be performed only once and the

factored matrix can be stored and used to obtain $\{\omega_b\}$ for different righthand side vectors, equation (95), during the iteration procedure (9).

(2). Computation of $\{u\}^{n+1}$ and $\{v\}^{n+1}$. The first N_b values of the vector $\{\omega\}$, representing the boundary vorticity, is updated using the new $\{\omega_b\}^{n+1}$ obtained in step (1). The rest of the vector components, representing ω at the interior points, is still given by the old values from the n^{th} iteration. The updated $\{\omega\}$ is then substituted in equations (71) and (72), and the new $\{u\}^{n+1}$ and $\{v\}^{n+1}$ are computed. The first N_b values of the vectors $\{u\}$ and $\{v\}$ represent the constant boundary values, and are not computed during the iteration. The computations are performed, explicitly, point by point.

(3). Computation of $\{h_b\}^{n+1}$. Applying equation (73) at the N_b boundary nodes, using the new $\{\omega_b\}^{n+1}$, $\{u\}^{n+1}$, $\{v\}^{n+1}$ gives a set of N_b algebraic equations containing the N_b components of $\{h_b\}^{n+1}$ as unknowns.

It is easy to show that if h_b is a constant, then the integral containing h_b vanishes. This means that the homogeneous set of equations, corresponding to the above set of equations, has a non-trivial solution. As a consequence, at most (N_b-1) equations of the set are independent of each other. In the computation procedure, it is permissible to assign an arbitrary value of h_b $(h = (\frac{V^2}{2} + \frac{p}{\rho})/\nu)$ at any one boundary node. This assignment is equivalent to specifying a reference pressure level. The solution of the problem, as far as the velocity, the vorticity, and the net force acting on closed bodies are concerned, is not affected by the specific pressure level used. When the boundary B is geometrically symmetric about a given axis and the boundary nodes are arranged symetrically, the rank of the set of equations was found to be (N_b-2). This additional

decrease in the number of independent equations can be eliminated by using a non-symmetric distribution of the boundary nodes, in which case the rank becomes (N_b-1). Another way to overcome this difficulty, is to use a known value of the pressure at one nodal point, e.g. as in known inflow boundary conditions. In the case of absence of this additional information, the value of p at this boundary node may be expressed as the average value at the two neighboring boundary nodes. In the last two alternatives, the number of unknowns becomes (N_b-2). The third choice is employed in the computer program.

Assuming the pressure p to vary linearly between the three boundary nodes N_b, N_b-1 and N_b-2, and taking p_{N_b} to be zero, gives

$$p_{N_b-1} = HBF \times p_{N_b-2} \tag{96}$$

where HBF is an interpolation parameter, given by

$$HBF = l_{N_b} / (l_{N_b} + l_{N_b-1}) \tag{97}$$

Hence, by definition,

$$h_{N_b} = \frac{\nu}{2} V^2_{N_b} \tag{98}$$

and,

$$h_{N_b-1} = \frac{\nu}{2}\left[V^2_{(N_b-1)} - HBF \times V^2_{(N_b-2)}\right] + HBF \times h_{N_b-2} \tag{99}$$

(4) Computation of $\{\omega\}^{n+1}$ at the interior nodes. The values of $\{\omega_b\}^{n+1}$, $\{u\}^{n+1}$, $\{v\}^{n+1}$, and $\{h_b\}^{n+1}$ are substituted into equation (73) and the resulting equation is evaluated to give ω at the interior nodes. In this step, old values from the previous iteration are used for values of ω in the right-hand side of equation (73). New values are used as soon as they are obtained.

Using the above iteration loop, it was found necessary to employ a point under-relaxation technique to obtain convergent solutions. The new value of ω^{n+1}, at any interior point, is computed from

$$\omega^{n+1} = ur \ \bar{\omega} + (1-ur)\ \omega^{n} \tag{100}$$

where ur is the under-relaxation parameter, $0. < ur < 1.$, and $\bar{\omega}$ is the value of ω computed in step (4) in the above mentioned iteration loop. The under-relaxation procedure is performed point by point and each component of $\{\omega\}$ is updated immediately after computing its new value from equation (100).

The rate of convergence was found to be best for some optimum value of the under-relaxation parameter, denoted by ur_o. The number of iterations required to obtain convergence, for a specified accuarcy, was found to increase for $ur < ur_o$, and convergence may not occur for values of ur much larger than ur_o. The value of this ur_o depends on the particular problem to be solved and on the Reynolds number. It was also found that, generally, $0. < ur_o \leq .5$. The rule is to try smaller values of ur, whenever divergence occurs.

The use of under-relaxation parameters is a common feature in most of the existing methods for solving the steady flow equations.

The final iteration convergence may be decided by one of the well known fractional-change criterions. The convergence may be assumed to occur when

$$\max_i \left| \omega_i^{n+1} - \omega_i^{n} \right| \Big/ \max_i \left| \omega_i^{n+1} \right| \leq \varepsilon \tag{101}$$

Usually, ϵ is set in the range of 10^{-4}-10^{-5}. Smaller values, of course, result in excessive computer time. The above criterion is employed in the computer program at the end of this report.

6. Illustrative Examples

1- Couette Flow

Steady Couette flows between two parallel infinite plates, with the upper one moving at a finite speed and the lower one stationary, subject to various pressure gradients in the direction of motion, were studied using the present solution procedure. This problem is one-dimensional and possesses a well known exact solution. In the present study, however, this problem is treated as a two-dimensional flow inside a square region. Velocity boundary conditions, on the square boundary, are prescribed in accordance with the exact solution. Pressure gradients are not explicitly prescribed, rather, their effects on the flow enter indirectly through the velocity boundary condition. The obtained numerical solutions are in excellent agreement with the exact solutions as shown in Fig. (6). The grid nodes were those defined by an equally spaced (8X8) rectanglar mesh.

All the computations, in this report, were done on a CDC Cyber-74 Computer system. The computation time used for the Couette flow problem was a few seconds per case.

2- Two-dimensional Straight-channel Flow

This problem is similar to the Couette flow problem. The only difference is that, in the channel flow case, both plates are stationary. The channel flow is, also, one dimensional and has an analytical solution. It was

treated, similar to the Couette flow, as a two-dimensional problem with prescribed velocity boundary conditions. The results were in excellent agreement with the analytical solution. The computation time was a few seconds per case.

3- Closed Cavity Flow

Flow inside a square cavity was studied for many Reynolds numbers, up to a thousand. The flow inside the cavity is induced by moving the upper lid at a constant speed. This problem has been studied by several authors, as a classical example for two-dimensional recirculating flows [10], and as a test case in developing numerical procedure for solving these types of flows, e.g. [10, 11]. The computed streamlines and vorticity contours, for Reynolds number of 100, are depicted in Figure (7). Figure (8) shows a comparison of predictions from the present method and that of reference [10] of the u-velocity component (parallel to the driving lid) along the vertical centerline of the cavity for Re = 100.

The present results agree favorably with predictions from Reference 10. The number of grid points was 169, and the vertices of the elements were chosen to coincide with the nodes of a (11X11) non-uniform Cartesian grid. The iteration was stopped using the criterion (101), with $\epsilon = 10^{-5}$. The computer time used to obtain all geometric coefficient matrices was 9 seconds, and the execution time for the iterative solution was 6 seconds. The geometric coefficient matrices depend, as explained in section (4), only on the grid distribution. They were computed once and used to obtain solutions at different Reynolds numbers.

The above three examples represent solutions of the flow equations in a square region; and differ only in the velocity boundary conditions.

Therefore, if the same grid system is used in the three cases, then the geometric coefficient matrices become the same for all of them. Besides, solutions at higher Reynolds numbers, for the same problem, can be started using the previously obtained solutions for a lower Reynolds number. These desirable features, of the current solution procedure, help to drastically reduce the needed computer time to generate these solutions.

4. Channel With a Square Cavity.

The flow configuration for this example is depicted in Figure (9). An open cavity, square in shape, with a moving plate located 1/6 the cavity width above the cavity, is studied using the present technique. The same configuration was investigated by Taylor and Hood (12), using Neumann boundary conditions at the entrance and exit sections.

The present velocity boundary conditions includes the no-slip condition on solid boundaries and linear entrance and exit velocity profiles, 1/3 cavity width upstream and downstream of the cavity. The vertices of the finite element mesh were those points, depicted in Figure (8), where velocity values are given. The number of points in the flow field was 97.

Results were obtained for Reynolds numbers up to 3000, although the flow is not expected to stay laminar up to this Reynolds number. The velocity field for a Reynolds number of 600 is shown in Figure (9). The present results show the same general pattern as in reference (12). The computer time required to compute the geometric matrices was about 7 seconds, and the time required for the iterative solution, starting from zero values for the vorticity at the interior nodes, was about 8 seconds for a Reynolds number of 1000.

5. Constricting Channel Flow

The Channel configuration is shown in Figure (10). The velocity boundary conditions at the exit and entrance sections are parabolic, corresponding to the fully-developed Poiseuille flow.

Computations were carried out for Reynolds numbers ranging from zero to 25. As the Reynolds number is increased from zero, the flow becomes non-symmetric about the constriction centerline, and a standing separation eddy develops downstream the constriction mid-section.

Figure (10) shows the computed streamlines for a Reynolds number of 25. The separation bubble started to occur, in the present computations, as the Reynolds number exceeded the value of 11. The vorticity distribution along the boundary wall, which is directly proportional to the wall shear stress, is shown in Figure (11) for the Reynolds number of 5. The present results are in agreement with the predicitons of reference [13], for the same configuration.

The problem is symmetric about the channel centerline. In the present computations, however, the symmetry condition was not fully utilized. The solution region included the upper and the lower halfs of the channel, and zero-velocity boundary conditions were specified on the upper and the lower walls of the channel. The grid nodes in one half were the images, with respect to the channel centerline, of the nodes in the other half. The geometric coefficients of a field variable at a node in the upper half and its image, in the lower half, are computed, then, the geometric coefficient corresponding to the later node is added to, or subtracted from, the geometric coefficient of the former node, depending on whether the field variable is symmetric or antisymmetric, respectively, with respect to the centerline. Hence, the actual computations, in the iteration

procedure, are performed only at the nodes in the upper half of the channel.

The number of nodes and the number of elements in each half were 329 and 563, respectively. The computer time used to compute the coefficient matrices was about two minutes, and the time to obtain a convergent solution was about one minute.

6. Turbulent Channel Flow

The integral representations method has recently been extended for steady turbulent flows [6]. The important advantages previously demonstrated for laminar flows are retained in that extension. The equations are mathematically "closed" by "modelling" the Reynolds stresses in terms of the mean rate of strain, through an eddy viscosity ν_t [15], as follows

$$\overline{v_i' v_j'} = \nu_t \left(\frac{\partial v_i}{\partial x_j} + \frac{\partial v_j}{\partial x_i} \right)$$

The eddy viscosity ν_t, is then expressed in terms of the turbulent kinetic energy (k), the rate of dissipation of turbulence energy (ϵ), and an additional numerical constant C, in the form

$$\nu_t = C \frac{k^2}{\epsilon}$$

The quantities k and ϵ are governed by differential transport equations which contains no additional dependent variables. The extended integral representations, along with the k-ϵ equations are solved simultaneously for the mean velocity, mean vorticity, k and ϵ.

The details of the extension of the integral representations methods to turbulent flows are presented in reference [6]. The turbulence modeling technique, used in the present study is described in reference [15].

In this section, we present only the results obtained for the problem of fully-developed turbulent channel flow. The problem was treated as a two-dimensional flow in a square region, in a manner similar to the above mentioned laminar case. The profiles of the mean velocity, Reynolds stress, ν_t , k and ϵ are depicted in figure (12), along with the experimental results of reference [16] .

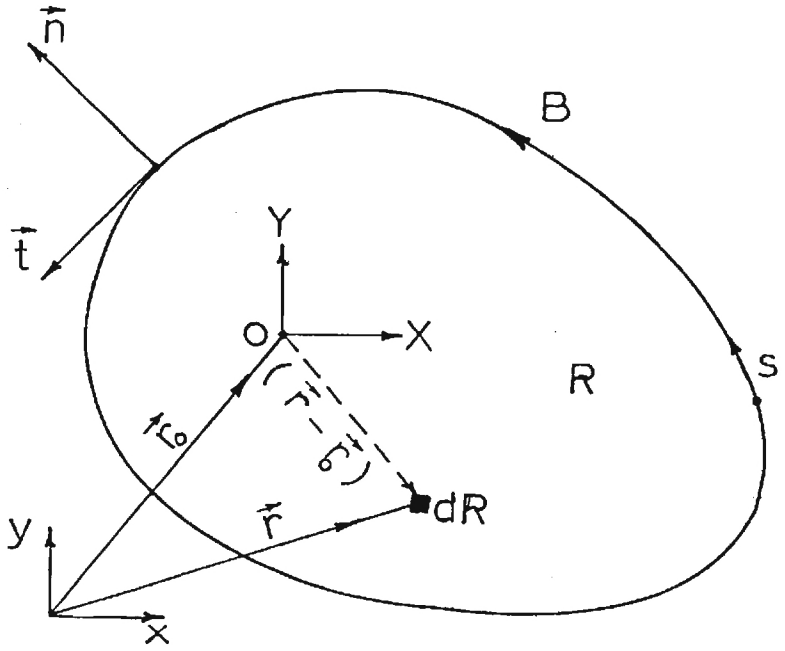


Figure 1. Flow domain and notations for 2-D flows

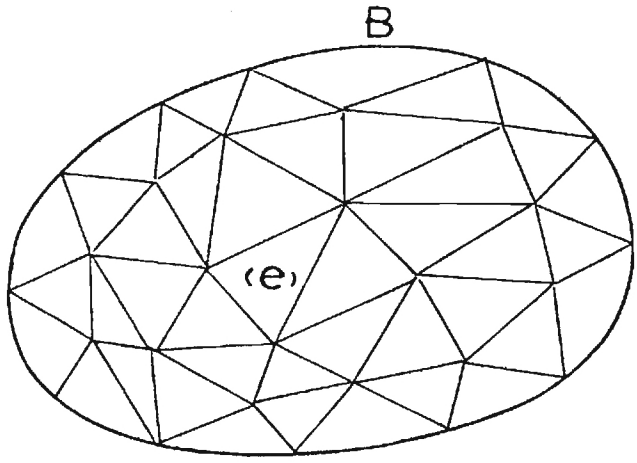


Figure 2. Division of a domain into finite elements

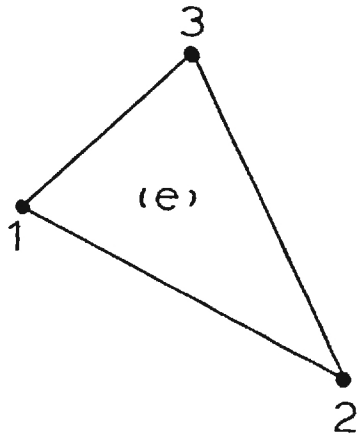


Figure 3. A typical triangular element

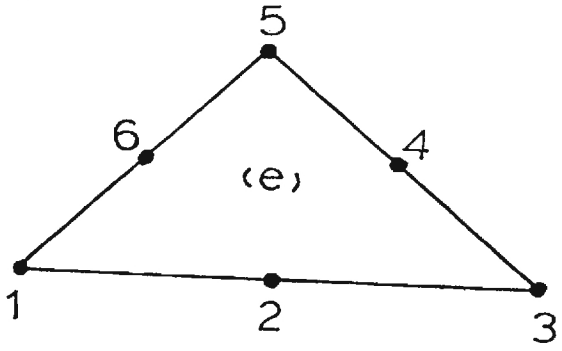
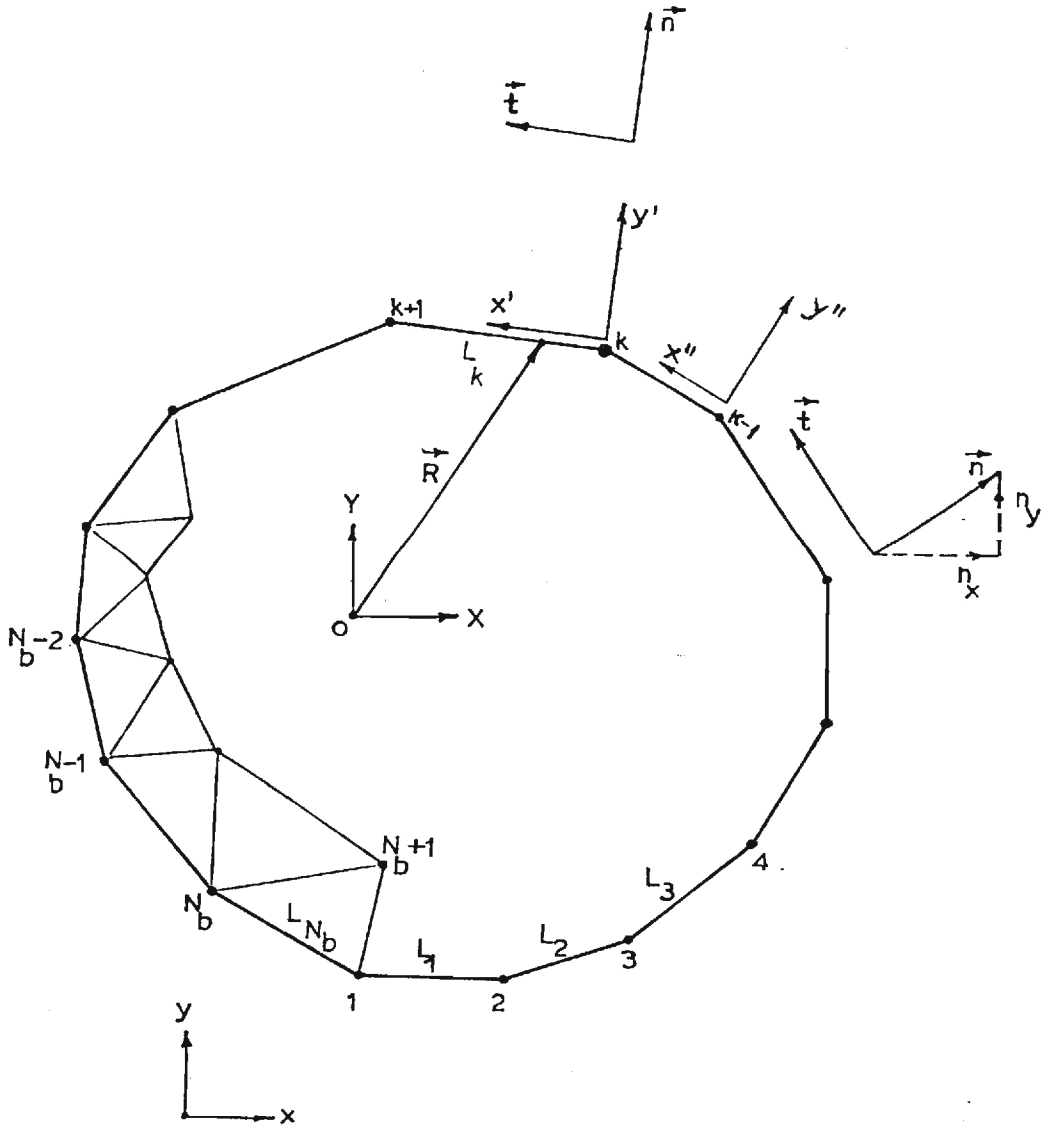


Figure 4. Triangular element with six nodes



Grid points index: from 1 to N_b Boundary nodes

from N_b + 1 to N interior nodes

Figure 5. Geometry and notation for boundary integrals

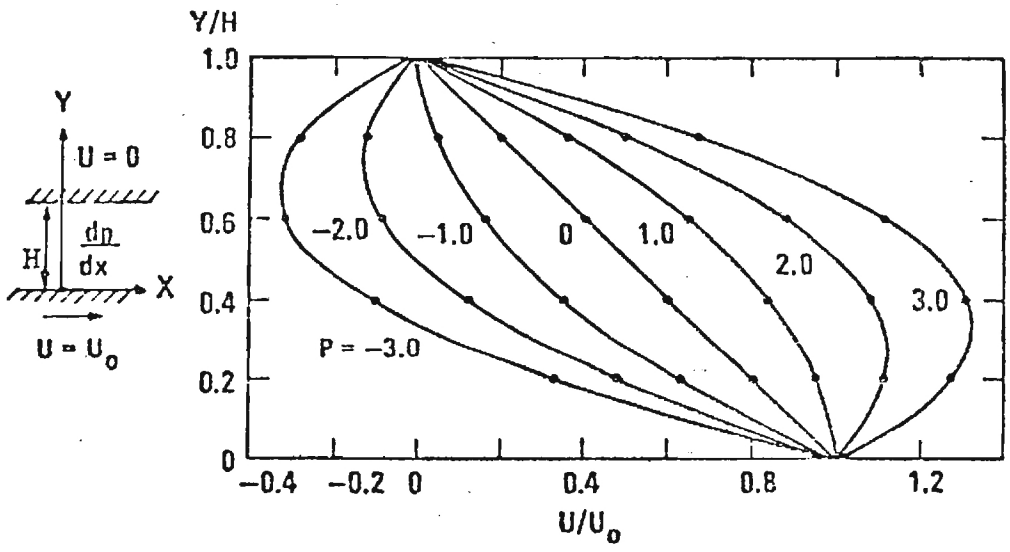
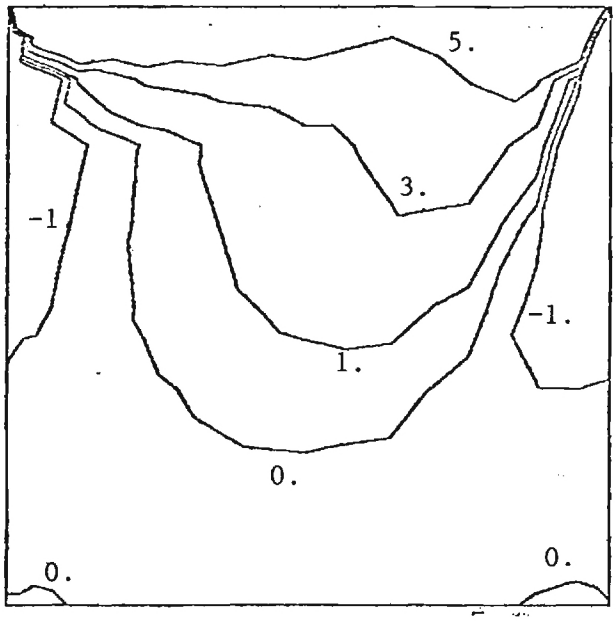


Figure (6) Couette Flow, velocity Profile

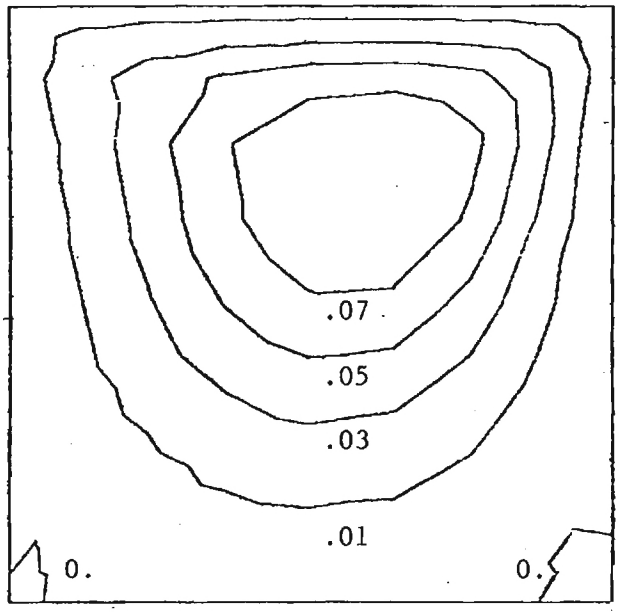
———— Exact

· Present results

$$P = \frac{H}{2\,U_o}\,\frac{d}{dx}$$



a - Vorticity contours



b - Streamlines

Figure 7. Closed cavity Flow, Re = 100.

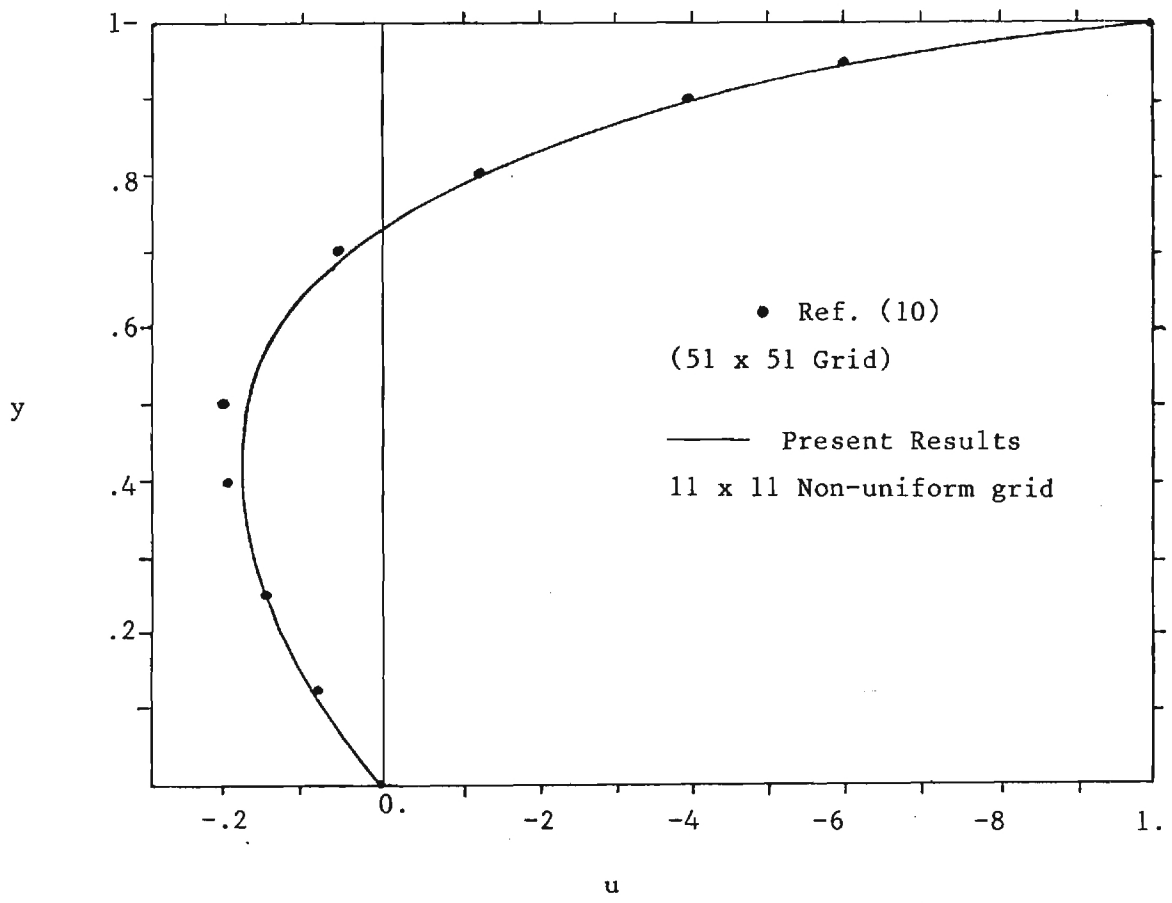


Figure 8. Velocity profile across the vertical centerline of the square cavity, Re = 100.

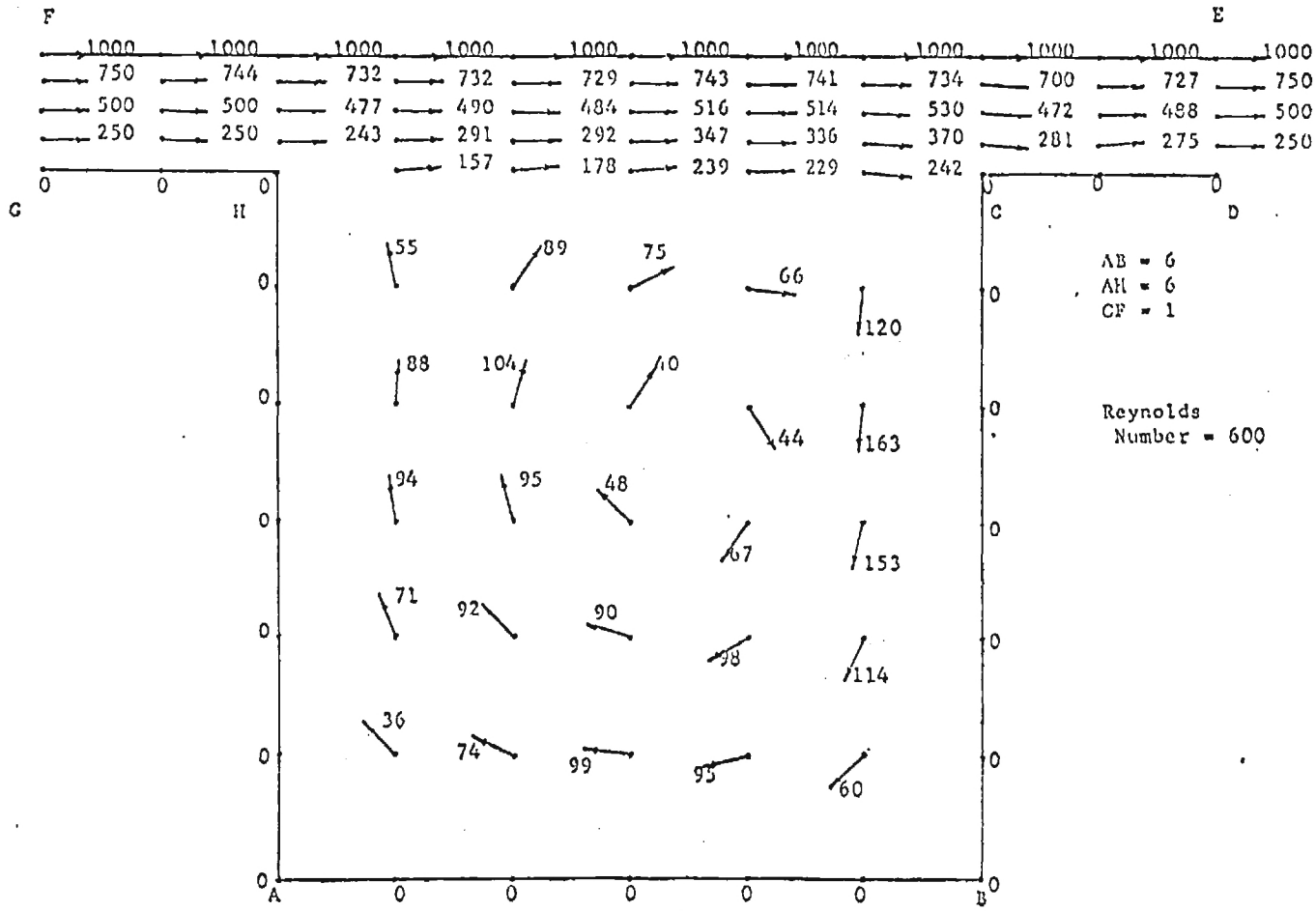
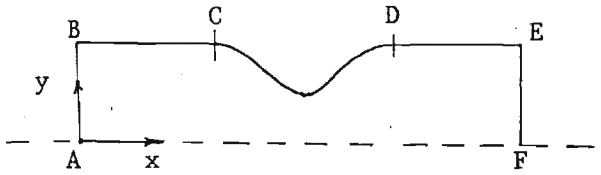


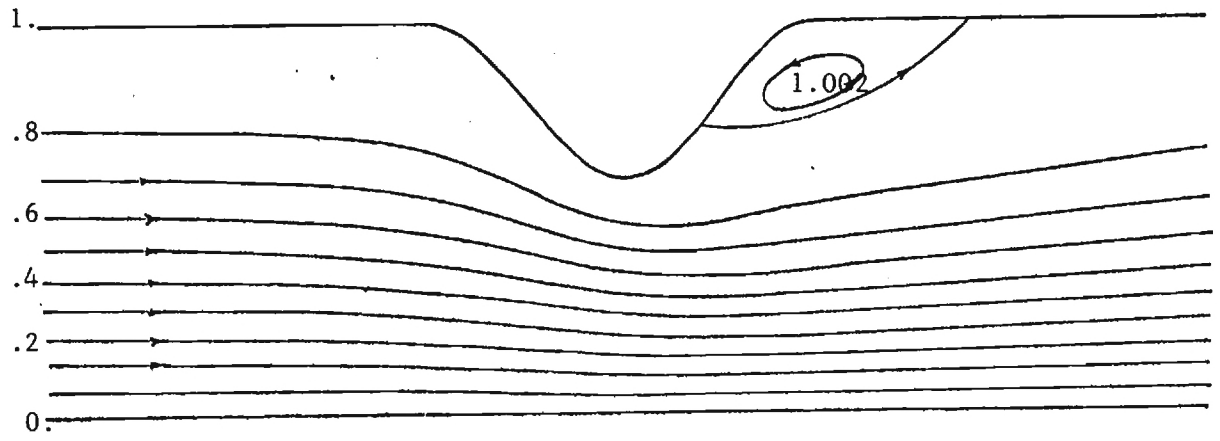
Fig. (9) Flow past an open cavity

AB = BC = EF = 1. , DE = 3.5

CD: y = 1. -.2 1.-cos2 (x-1.)



a. Coordinates and geometry of a constricting channel



b. Streamlines

Figure 10. Flow inside a constricted channel, Re = 25.

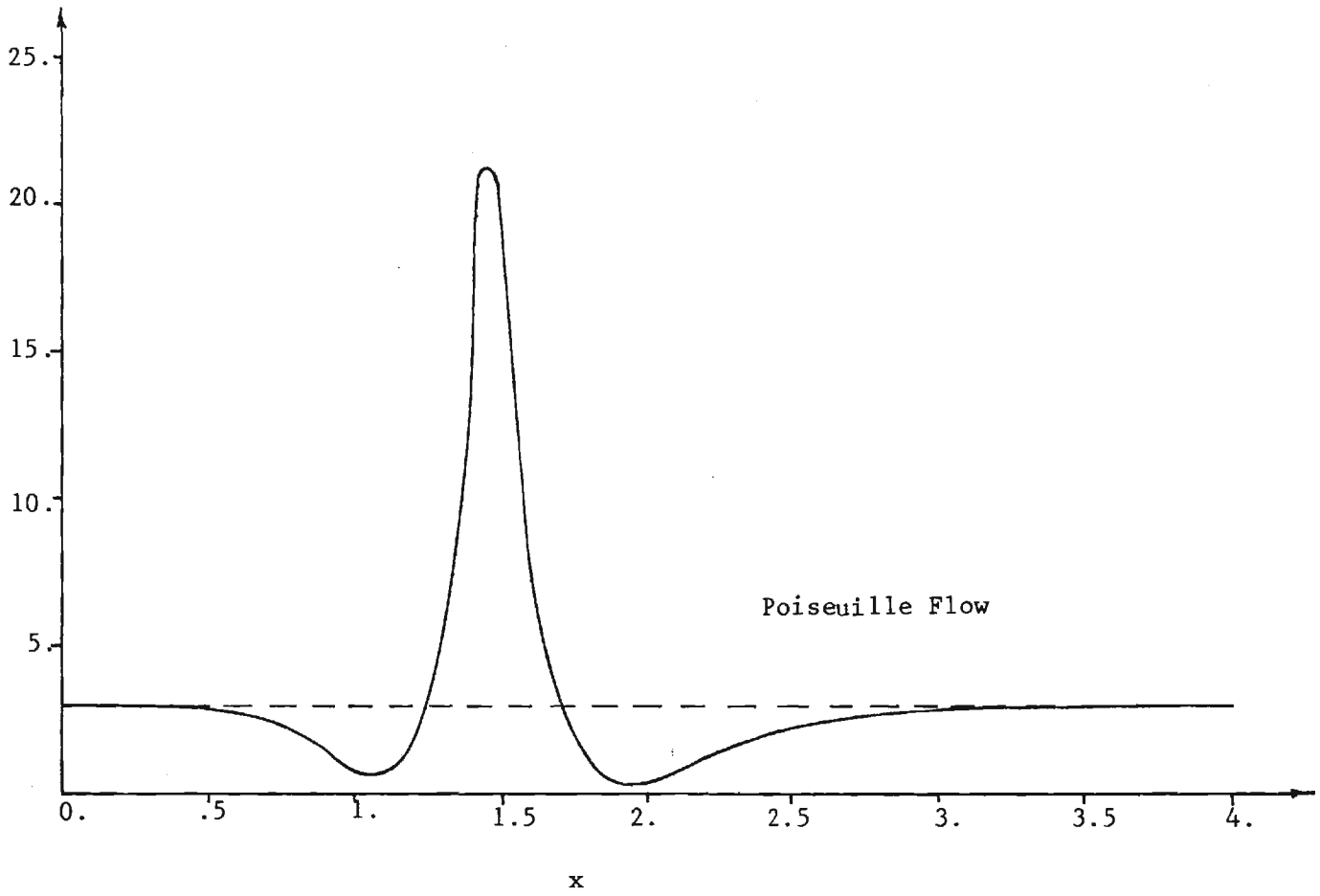
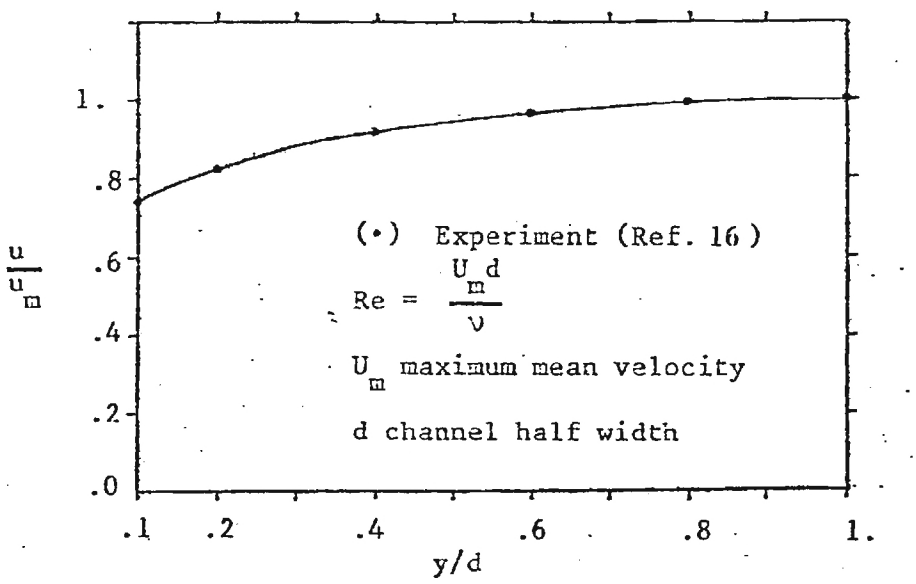
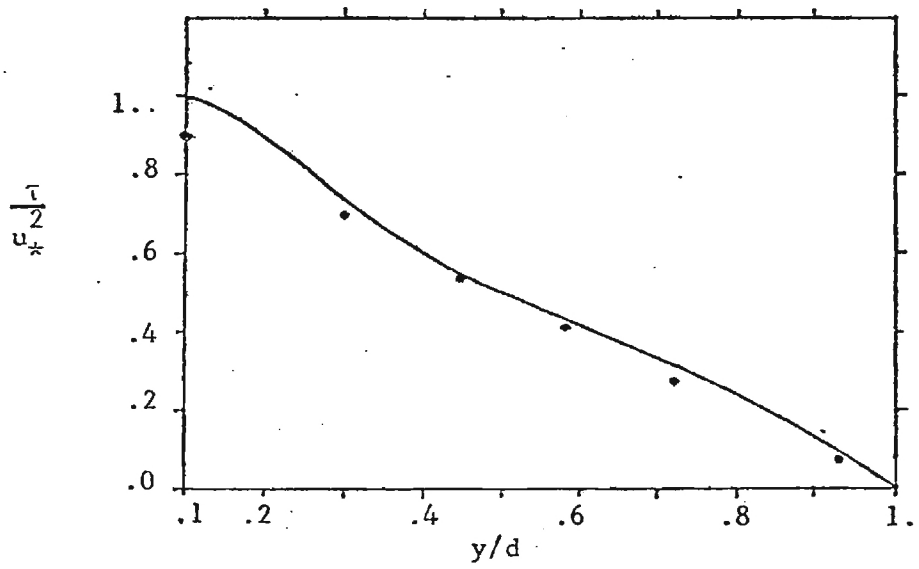


Figure (11) Flow inside a constricted channel; wall vorticity profile, Re = 25.

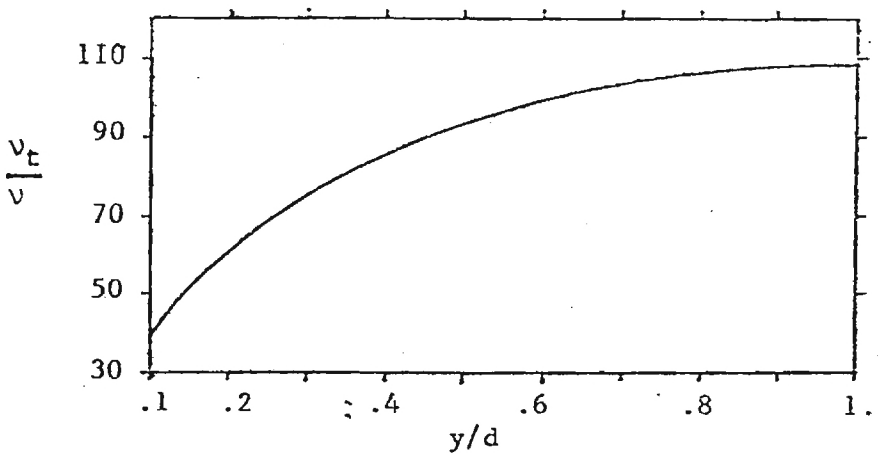


(a) Mean Velocity Profile

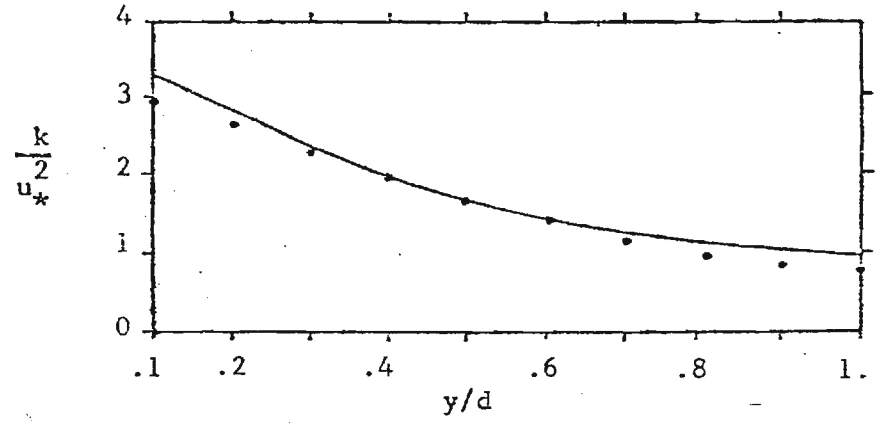


(b) Reynolds Stress

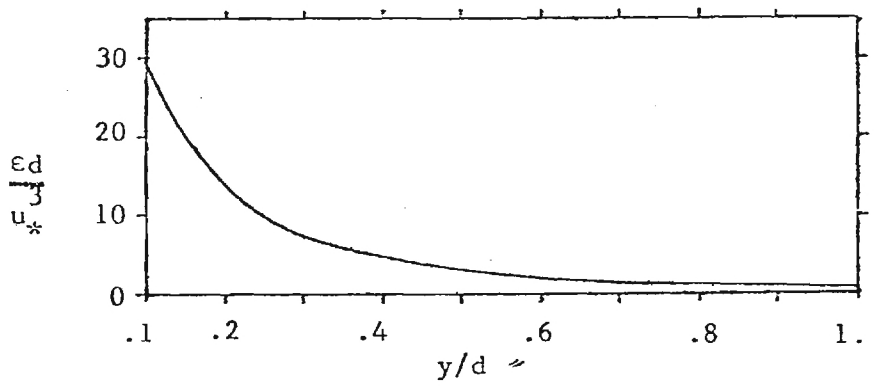
Figure 12. Turbulent Channel Flow (Re = 30,800)



(c) Turbulent eddy viscosity



(d) Turbulent Kinetic Energy



(e) Turbulent Energy Dissipation Rate

Figure 12. (continued)

Appendix A

Numerical Evaluation of $\iint_R f(x,y) \frac{x^k y^l}{x^2 + y^2} dx\, dy$ over Arbitrary Domains using Finite Elements

A.1 Introduction

In the present section, a semi-analytical method for evaluating the class of integrals

$$I = \iint_R f(x,y) \frac{x^k y^l}{x^2 + y^2} dx\, dy \qquad \text{(A-1)}$$

with k and l are positive integers, is presented. The function f(x,y) can be any reasonably well-behaved function, defined over the domain R which is bounded by an arbitrary curve B, Figure (2). This class of integrals are, sometimes, associated with the use of integral methods employing Green's functions, in two dimensions, and with the theory of potential. The method yields the exact value of I for polygonal regions and f being a polynomial in x and y .

A.2 Evaluation of I using Finite Elements

The domain R is mapped into finite elements, any of which may, in general, be a polygon of S sides. The above integral can be expressed as the sum of the cotributions from individual elements, designated by

(e), as follows,

$$I = \sum_e \iint_{(e)} f^{(e)} \frac{x^k y^l}{x^2 + y^2} \, dxdy \qquad (A\text{-}2)$$

Introducing the element interpolation functions expressing $f^{(e)}$, for each element (e), as a double polynomial in x and y of the form

$$f^{(e)} = \sum_{p,q} K_{p,q} \, (x^p \, y^q) \qquad (A\text{-}3)$$

equation (A-2) becomes,

$$I = \sum_e \sum_{m,n} K_{p,q} \, I^{(e)}_{m,n} \qquad (A\text{-}4)$$

where

$$I^{(e)}_{m,n} = \iint_{(e)} \frac{x^m y^n}{x^2 + y^2} \, dxdy \qquad (A\text{-}5)$$

with

$$\begin{aligned} m &= k + p \\ n &= l + q \end{aligned} \qquad (A\text{-}6)$$

Each element (e) is associated with a certain number of nodes. These nodes may be located at the polygon boundary, polygon vertices, or inside the element. The number of the elementary interpolation functions $(x^p y^q)$, introduced in an element, should be equal to the number of nodes associated with this element. The constants $K_{p,q}$, are obtained by satisfying equation (A-3) at the element nodes and solving for $K_{p,q}$ in terms of $f^{(e)}$ at these nodes. The completeness, and the computational efficiency of different interpolation functions, associated with different elements, are discussed in reference [8].

The number, shape, and distribution the finite elements, depend on the required accuracy and the order of interpolation used.

A.3 Evaluation of $I^{(e)}_{m,n}$

For simplicity, we first consider a triangular element (Δ). The vertices are numbered, sequentially, in the counterclockwise direction, with the vertex 4 is the same as vertex 1, Figure (A.1). The line joining the two vertices i and (i+1), (i = 1,2,3), is denoted by L_i, and the triangle formed by L_i and the origin o is denoted by Δ_i.. Hence, the integral over (Δ) can be expressed in terms of integrals over Δ_1, Δ_2, and Δ_3, as follows

$$I^{(\Delta)}_{m,n} = I^{\Delta_2}_{m,n} + I^{\Delta_3}_{m,n} - I^{\Delta_1}_{m,n} \qquad \text{(A-7)}$$

The integrals I^{Δ_i} can be best evaluated by using the polar coordinates, defined by

$$\begin{aligned} x &= r \cos \Theta \\ y &= r \sin \Theta \end{aligned} \qquad \text{(A-8)}$$

The equation of the line L_i, in terms of the x – intercept a_i, and the y – intercept b_i, is given by

$$r_i = D_i / \sin (\Theta + \phi_i) \qquad \text{(A-9)}$$

where

$$\begin{aligned} D_i &= a_i b_i / c_i \\ \sin \phi_i &= b_i / c_i \\ \cos \phi_i &= a_i / c_i \end{aligned} \qquad \text{(A-10)}$$

with

$$c_i = (a_i^2 + b_i^2)^{1/2} \qquad \text{(A-11)}$$

Hence

$$I_{m,n}^{(\Delta_i)} = \int_{\theta_i}^{\theta_{i+1}} d\theta \cos^m\theta \sin^n\theta \int_o^{Ai/\sin(\theta+\phi_i)} r^{m+n-1} dr \qquad (A\text{-}12)$$

where θ_i is the angular position of the vertex i. Substituting equation (A-12) into equation (A-7), after performing the r integration, and taking into consideration that θ increases in the counterclockwise direction, yield

$$I_{m,n}^{\Delta} = \sum_{i=1}^{3} \frac{A_i^{(m+n)}}{m+n} \int_{\theta_i}^{\theta_{i+1}} G_{m,n}\, d\theta \quad ;(m+n) \geq 1 \qquad (A\text{-}13)$$

where

$$G_{m,n} = \cos^m\theta \sin^n\theta / \sin^{(m+n)}(\theta + \phi_i) \qquad (A\text{-}14)$$

The integral $I_{m,n}^{\Delta}$ for the special case $m = n = 0$ does not exist when the origin (o) lies on the boundary of the element Δ or inside the element. If the origin (o) is outside the element Δ, then

$$I_{o,o}^{\Delta} = \sum_{i=1}^{3} \left[(\theta_{i+1} - \theta_i) \ln Ai - G(\theta_{i+1} + \phi_i) + G(\theta_i + \phi_i) \right] \qquad (A\text{-}15)$$

with

$$G(x) = x \ln x - x - \frac{x^3}{18} - \frac{x^5}{900} - \ldots - \frac{2^{2n-1} B_n x^{2n+1}}{n(2n+1)!} \qquad (A\text{-}16)$$

where B_n are Bernoulli coefficients [14]. This singular case is of less practical importance than the case of $(n+m) \geq 1$, and is not discussed further. As an example, in solving steady flow problems using integral representations [4], $(m+n)$ is always ≥ 1.

Similarly, for a polygon of S sides, equation (A-13) becomes,

$$I_{m,n} = \sum_{i=1}^{S} \frac{Ai^{(m+n)}}{(m+n)} \int_{\theta_i}^{\theta_{i+1}} G_{m,n}\, d\theta \ldots (m+n \geq 1) \qquad (A\text{-}17)$$

The above integral can be easily carried out using the transformation ($\alpha = \Theta + \phi_i$) in $G_{m,n}$, then

$$G_{m,n} = \cos^m (\alpha - \phi_i) \sin^n (\alpha - \phi_i) \; / \sin^{(m+n)} \alpha$$

$$= (\cos \phi_i \cdot \cot \alpha + \sin \phi_i)^m \; (\cos \phi_i - \sin \phi_i \cdot \cot\alpha)^n \qquad \text{(A-18)}$$

Using the binomial formula to expand each bracket in equation (A-18), $G_{m,n}$ can be expressed in a power series expansion in $\cot \alpha$, as follows

$$G_{m,n} = \sum_{l=o}^{m+n} B_l \cot^l \alpha \qquad \text{(A-19)}$$

where

$$B_l = \sum_{j=o}^{l} (-)^{l-j} C_j^m C_{l-j}^n (\cos \phi_i)^{n+2j-l} \; (\sin \phi_i)^{m+l-2j} \qquad \text{(A-20)}$$

and

$$C_j^m = \frac{m!}{j!\,(m-j)!} \qquad \text{(A-21)}$$

The quantity C_j^m is taken to be zero for $j > m$

Substituting equation (A-19) into equation (A-17), and performing the integrals term by term, gives

$$I_{m,n} = \sum_{i=1}^{S} \frac{A_i^{m+n}}{m+n} \cdot \left(\sum_{l=o}^{m+n} B_l \, F_{l+1} \right) \qquad \text{(A-22)}$$

where the quantities F_l are calculated from the following recurrence relations

$$F_1 = \Theta_{i+1} - \Theta_i$$

$$F_2 = \ln \left| \sin (\Theta_{i+1} + \phi_i) \; / \; \sin (\Theta_i + \phi_i) \right|$$

and

$$F_l = \frac{-1}{l-1}\left\{\cot^{l-1}(\theta_{i+1}+\phi_i) - \cot^{l-1}(\theta_i+\phi_i)\right\} - F_{l-2} \quad \ldots \quad (l \geq 3) \qquad \text{(A-23)}$$

It should be noted that when the domain R is a polygon of any number of sides and $\omega(x,y)$, equation (A-1), is constant over R, then the exact value of I is given by equation (A-22).

The function $\sin(\theta_i+\phi_i)$ can be computed, efficiently, from the relation

$$\sin(\theta_i+\phi_i) = \sin\theta_i \cos\phi_i + \cos\theta_i \sin\phi_i \qquad \text{(A-24)}$$

where

$$\sin\theta_i = \frac{y_i}{x_i^2 + y_i^2} \qquad \cos\theta_i = \frac{x_i}{x_i^2 + y_i^2} \qquad \text{(A-25)}$$

and $\cos\phi_i$ and $\sin\phi_i$ are given by equation (A-10). The same technique is used to compute $\cos(\theta_i+\phi_i)$ and hence, $\cot(\theta_i+\phi_i)$. This technique avoids the direct computation of the corresponding trigonometric functions, which is a time-consuming process in computers. It should also be noted that if $(\theta_i+\phi_i)$ is equal to zero, then $(\theta_{i+1}+\phi_i)$ also vanishes, resulting in an indefinite value of F_2, in equation (A-23). In this case, the corresponding line L_i passes through the origin and its contribution to $I_{m,n}$ is equal to zero.

A few comments on using the polar coordinates transformation, equations (A-8), to evaluate these integrals are deemed appropriate. The transformed integral is equivalent to the original one, only if we define a branch cut in the (x-y) plane, excluding the origin. The domain of integration should not cross that branch cut. Thus, in evaluating the integrals I^{Δ_i}, we define θ to be such that $0 \leq \theta < 2\pi$, unless the x-axis passes

through the triangle Δ_i, in which case we define θ to be $-\pi \leq \theta < \pi$. This procedure determines the principal values of θ_i and θ_{i+1} to be used in evaluating F_1 in equation (A-23), since these angles are determined, numerically, through some trigonometric functions such as those given by equation (A-25). More conveniently, the quantity F_1, instead of the individual values of θ_i and θ_{i+1}, can be computed as the angle between the two radius vectors $\vec{r}_1 = (x_i, y_i)$ and $\vec{r}_2 = (x_{i+1}, y_{i+1})$, with $o \leq |F_1| \leq \pi$. The absolute value of F_1 can be obtained through the dot-product of these two vectors, as follows,

$$|F_1| = | \cos^{-1} \{ (x_{i+1} x_i + y_{i+1} y_i) /(r_i r_{i+1})\}| \qquad \text{(A-26)}$$

where

$$r_i = (x_i^2 + y_i^2)^{1/2}$$
$$r_{i+1} = (x_{i+1}^2 + y_{i+1}^2)^{1/2} \qquad \text{(A-27)}$$

The principal branch of the function $\cos^{-1}$, in equation (A-26) is $\{0, \pi\}$, which is usually supplied on computers.

The sign of F_1 is positive if the rotation from $\vec{r}_i$ to $\vec{r}_{i+1}$, through the principal angle $|F_1|$, is counterclockwise, and the sign is negative if the rotation is clockwise. In other words, the sign of F_1 is given by the sign of the vector cross-product $(\vec{r}_1 x \vec{r}_2)$, hence,

(A-28)

$$\text{sign } (F_1) = \text{sign } (y_{i+1} x_i - x_{i+1} y_i)$$

The cross-product $(\vec{r}_1 x \vec{r}_2)$ vanishes when the line L_i passes through the origin; in which case the contribution from that line to $I_{m,n}$ vanishes,

and need not be considered in the computation.

Finally, it is easy to show that the quantity F_2, equation (A-23), can be expressed as follows,

$$F_2 = \ln (r_i / r_{i+1}) \qquad \text{(A-29)}$$

The main objective behind the above manipulation, is to minimize the use of transcendental functions, therefore, reducing the required computer time for the evaluation of these integrals.

A.4 A Computer Subroutine for Computing the Integrals $I^{(e)}_{m,n}$

In the present section, an algorithm for the efficient computation of these integrals is outlined. The corresponding computer code (Subroutine GF), in Fortran IV computer language, is provided in section A.5.

The computation of any member of this class of integrals depends on A_i, $\sin\phi_i$, and $\cos\phi_i$, defined by equation (A-10), and on F_1 and F_2, given by equation (A-23). These quantities are common in all members, hence, it is more economical to evaluate all the needed members of this class of integrals, over the same element, simultaneously.

It is more efficient, particularly for small (m+n), to obtain the coefficients B_1, equation (A-20), by hand and supply the answer to the computer subroutine. This process becomes more difficult to carry out as 1 gets large, in which case the computation of B_1 is programmed too. The values of B_1 required in computing $I_{m,n}$, for $1 \leq (m+n) \leq 4$, are given in table (A-1)

It should be noted that some members, of this class of integrals, can be obtained in terms of others plus additional simpler integrals. For example, it can be easily shown that

$$I_{2,o} = \text{AREA} - I_{o,2} \qquad \text{(A-30)}$$

where AREA is the area of the element.

In the computer subroutine, the integrals $I_{1,0}$, $I_{0,1}$, $I_{1,1}$ and $I_{0,2}$ needed in the present analysis, are computed using the explicit values of the coefficients B_1, table A.1. The computation of any additional integral can be easily done by inserting few cards to compute the corresponding B_1 and F_1. The integrals are assumed to be in the more general form

$$I_{m,n} = \iint \frac{(x-x_o)^m \ (y-y_o)^n}{(x-x_o)^2 + (y-y_o)^2} \ dxdy \qquad \text{(A-31)}$$

where (x_o, y_o) is an arbitrary point.

The procedure consists of the following main steps

1- The integral is obtained as the sum of individual contributions from the S sides forming the polygon -- loop (10).

2- Determination of the coordinate relative to the point (x_o, y_o).

3- Computing the x - intercept and the y-- intercept of the line L_i.

At this point, some checks should be done to avoid computing infinite and indefinite quantities. Besides, the contribution from the line L_i is not considered when the line passes through the point (x_o, y_o). This step is illustrated by the flow chart in Figure A.2.

4- Evaluation of D_i, $\sin\phi_i$, and $\cos\phi_i$, either from equation (A-10), or using their limiting values, as illustrated in the flow chart.

5- Computing the functions F_1, F_2, and F_3, equation (A-23), and, hence, the line contribution.

The main Fortran variables are explained below

Fortran name	Description
SF(I)	The integrals $I^{\Delta}_{m,n}$, with SF(1) = $I_{1,0}$, SF(2) = $I_{0,1}$, SF(3) = $I_{1,1}$, and SF(4) = $I_{0,2}$. The integral $I_{2,0}$(SF(5)) is computed, in the main program, from equation (A-30).
XE(I), YE(I)	Coordinates of the element vertices. The nodes are numbered sequentially, with the index I increasing form 1 to (S+1) as we go around the element in the counterclockwise direction. The points (S+1) and (1) are coincident.
DELTA	Small number depending on the element size can be taken to be 10^{-3} l, where l is the length of smallest side of the element.
PI	Slope of the line L_i
CX,CY	The x-intercept and the y-intercept of L_i
D	D_i, equation (A-10)
A	Cos ϕ_i, equation (A-10)
B	sin ϕ_i, equation (A-10)
RIS,RJS	r_i^2 and r_{i+1}^2, equation (A-27)
COTI,COTJ	cot $(\Theta_i+\phi_i)$ and cot $(\Theta_{i+1}+\phi_i)$
CIJ	Cos (F_1). equation (A-26)
F1,F2,F3	Functions defined by equations (A-23) and (A-29)

Table A.1 The Coefficients B_1 for $1 \leq (m+n) \leq 4$ $(1 \leq m+n)$

			$a=\cos \phi_i$		$b=\sin \phi_i$		
m+n	m	n	BO	B_1	B_2	B_3	B_4
1	0	1	a	-b	--	--	--
	1	0	b	a	--	--	--
2	0	1	a^2	-2ab	b^2	--	--
	1	1	ab	$1-2b^2$	-ab	--	--
	2	0	b^2	2ab	a^2	--	--
3	0	3	a^3	$-3a^2b$	$3ab^2$	$-b^3$	--
	1	2	a^2b	a^3-2ab^2	b^3-2a^2b	ab^2	--
	2	1	ab^2	$2a^2b-b^3$	a^3-2ab^2	$-a^2b$	--
	3	0	b^3	$3ab^2$	$3a^2b$	a^3	--
4	0	4	a^4	$-4a^3b$	$6a^2b^2$	$-4ab^3$	b^4
	1	3	a^3b	$a^2(a^2-3b^2)$	$-3ab(a^2-b^2)$	$b^2(3a^2-b^2)$	$-ab^3$
	2	2	a^2b^2	$-2ab(b^2-a^2)$	$b^4-4a^2b^2+a^4$	$2ab(b^2-a^2)$	a^2b^2
	3	1	ab^3	$b^2(3a^2-b^2)$	$3ab(a^2-b^2)$	$a^2(a^2-3b^2)$	$-a^3b$
	4	0	b^4	$4b^3a$	$6b^2a^2$	$4ba^3$	a^4

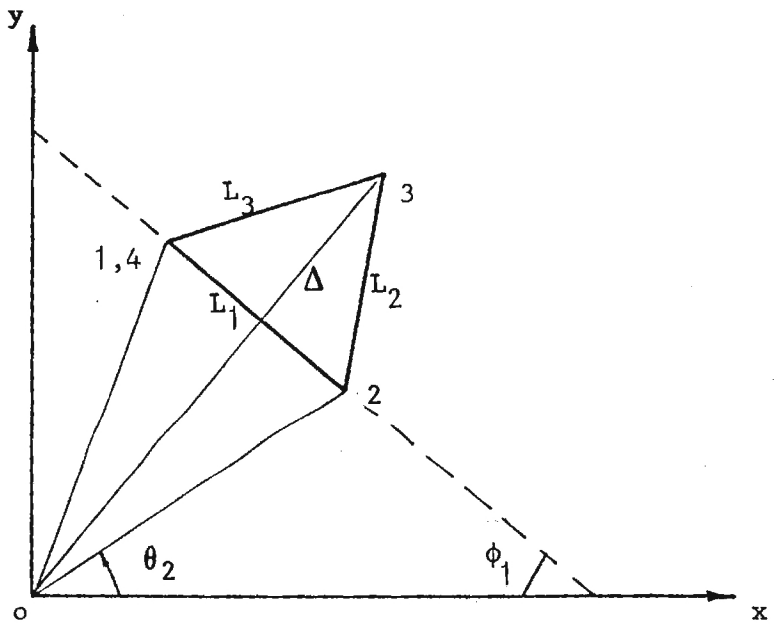


Figure A.1 Geometry and notation for a triangular element

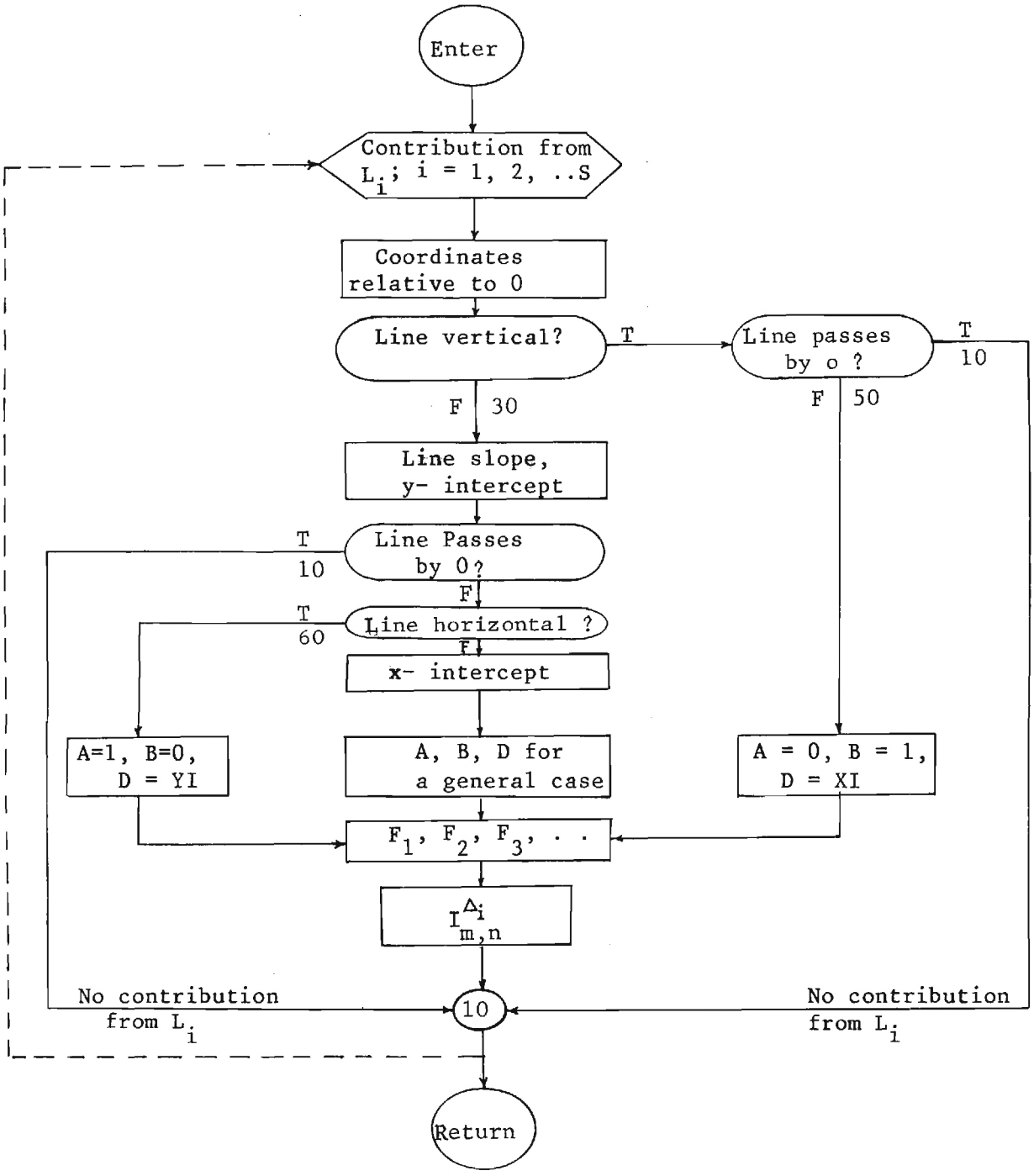


Figure (A.2) Flow-diagram for subroutine GF

Appendix B

Evaluation of the singular parts in the Boundary Integrals (section 3.2)

The quantity $G_{o,n}$, equation (51), represents the contributions, to the integral I_u, from the two line segments (n-1) and (n), and is given by

$$G_{o,n} = \left(\frac{y'_o}{l_n} f_1^n + \left(\frac{x'_o}{l_n} - 1\right) f_o^n\right) - \left(\frac{y''_o}{l_{n-1}} f_1^{n-1} + \frac{x''_o}{l_{n-1}} f_o^{n-1}\right) \qquad (B\text{-}1)$$

with

$$f_o^{n-1} = \tan^{-1}\frac{(l_{n-1} - x''_o)}{y''_o} + \tan^{-1}\frac{x''_o}{y''_o} \qquad (B\text{-}2)$$

$$f_o^n = \tan^{-1}\frac{(l_n - x'_o)}{y'_o} + \tan^{-1}\frac{x'_o}{y'_o} \qquad (B\text{-}3)$$

$$f_1^{n-1} = \frac{1}{2} \ln\left(\frac{(l_{n-1} - x''_o)^2 + y''^2_o}{x''^2_o + y''^2_o}\right) \qquad (B\text{-}4)$$

$$f_1^n = \frac{1}{2} \ln\left(\frac{(l_n - x'_o)^2 + y'^2_o}{x'^2_o + y'^2_o}\right) \qquad (B\text{-}5)$$

where (x'_o, y'_o) represent the coordinates of the point (o) with respect to the coordinate-system (x', y'), pertaining to the line L_n and with origin at the point (n), and (x_o'', y_o'') are the coordinates of (o) with respect to the axes (x",y"), pertaining to the line L_{n-1} and with origin at point (n-1), as depicted in Figure (B-1).

As the point o approaches the point n, we have

$$\begin{aligned} &x'_o \longrightarrow o. \;;\; y'_o \longrightarrow -o. \\ &x''_o \longrightarrow l_{n-1} \;;\; y''_o \longrightarrow -o. \qquad ;(x_o, y_o) \longrightarrow (x_n, y_n) \\ &y'_o f_1^n \longrightarrow o. \;;\; y''_o f_1 \longrightarrow o. \end{aligned} \qquad (B\text{-}6)$$

It should be noted that y_o' and y_o'' tend to -0, and not to +0, as (x_o, y_o) tends to (x_n, y_n), for internal flows inside the boundary B.

Substituting equations (B-6) into equation (B-1), yields

$$G_{o,n} = -\left(f_o^n + f_o^{n-1}\right)\Big|_{o \to n} \tag{B-7}$$

Referring to Figure (B.1), it can be easily shown that

$$f_o^n = -(\alpha_1 + \alpha_2)$$
$$f_o^{n-1} = -(\alpha_3 + \alpha_4)$$

where α_1, α_2, α_3, and α_4 are angles defined in Figure (B.1). Hence, it can be determined that $G_{o,n}$, as the point o coincides with the point n, is equal to the external angle between L_{n-1} and L_n. In other words, we can write

$$G_{n,n} = \pi + \beta \tag{B-8}$$

where β is the angle between the two vectors x'' and x' , Figure B.1, and with the principal value $o \leq |\beta| \leq \pi$. The sign of β is positive if the rotation from x'' to x', through the principal angle $|\beta|$, is counterclockwise, and is negative if the rotation is clockwise. A method for computing that angle is given at the end of Appendix A.

Similarly, as the point (o) coincides with the point n, the quantity, $H_{o,n,}$, equation (54), becomes

$$H_{n,n} = -\left(f_1^n + f_1^{n-1}\right)\Big|_{o \to n} \tag{B-9}$$

Referring to Figure (B.1), we can write equations (B-4) and (B 5) in the following form

$$f_1^{n-1} = \ln\left(\frac{r_n}{r_{n-1}}\right) \tag{B-10}$$

$$f_1^n = \ln\left(\frac{r_{n+1}}{r_n}\right) \tag{B-11}$$

where r_{n-1}, r_n, and r_{n+1} are the distances from the point (o) to the points (n-1), n, and (n+1), respectively.

Substituting equations (B-10) and (B-11) into equation (B-9), and performing the limit operation, gives

$$H_{n,n} = \ln (l_{n-1} / l_n) \qquad \text{(B-12)}$$

where l_{n-1} and l_n are the lengths of the line segments L_{n-1} and L_n, respectively.

In the computer program, the two matrices $[G_{m,n}]$ and $[H_{m,n}]$ are computed simultaneously, since their elements contain the same basic functions. The elements of the matrices are not computed directly from equations (83) and (84), rather, the integrals over the line segments are evaluated individually from equations (49) and (53-a), and the contribution to the corresponding elements are added to those elements. For example, the integration over the line segment L_n is evaluated, and the contributions to the elements (m,n-1) and (m,n) are added to those elements. When the point (m), at which the dependent variables u, v and ω are computed, lies on the boundary, the contribution from the two neighboring line segments are first neglected, and then added in a later step, as given below.

It is easy to show that, when the field point (o) coincides with the boundary point (n), the contributions to the integrals I_u and I_v from the two line segments L_{n-1} and L_n, are given by

$$I_u^{n-1} + I_u^n = G_{n,n}\, u_n \qquad \text{(B-13)}$$

$$I_v^{n-1} + I_v^n = v_{n-1} + H_{n,n}\, v_n - v_{n+1} \qquad \text{(B-14)}$$

where $G_{n,n}$ and $H_{n,n}$ are given by equations (B-8) and (B-12). Hence, the contributions of the (neglected) two line segments can be taken care of by adding $G_{n,n}$ to the element (n,n) of the matrix [G], and adding +1, $H_{n,n}$,-1 to the elements (n,n-1), (n,n) and (n,n+1) of the matrix [H], respectively. This modification is performed for all boundary points, $1 \leq n \leq N_b$.

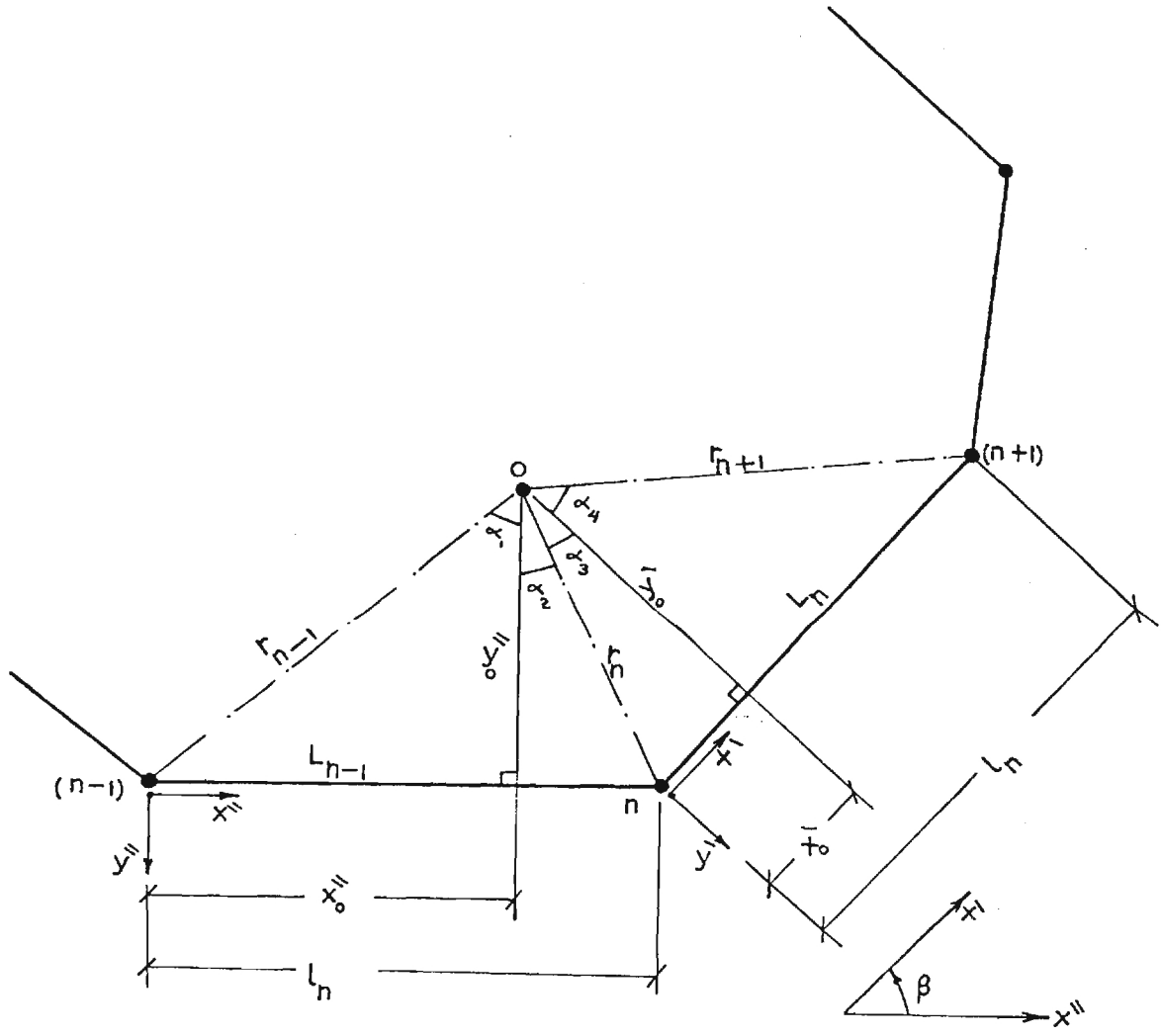


Figure B.1 Coordinates and geometric parameters for two successive boundary segments.

Appendix C The Computer Program

C.1 Introduction

In this appendix, a computer program for the computation of two-dimensional laminar flows, and based on the integral representation method, is presented. The algorithm is a direct implementation of the computational procedure described in section 5.

The main program is divided into several sections, for convenience of understanding and checking, as illustrated in the flow-diagram in Figure C.1.

The program should be supplied with, as input, the following information.

1-The finite element mapping of the solution region R, into triangular elements. This can be provided by specifying a two-dimensional array NE(I, J), giving the indices of the three nodes (J; J = 1, 2, 3) of each element (I; I = 1, 2, ...NFE, where NFE is the total number of elements in the region R) in the mapping.

2-The x and y-coordinates of each node in the grid system.

3-The velocity components at the boundary nodes

4-The Reynolds number, the maximum number of iterations allowed for the iteration process, the values of the parameter ϵ in the convergence criterion, and the value of the under relaxation parameter UR.

Given the above information, the program first computes all the geometric coefficients matrices, and then solves, iteratively, by going through the sequence of steps described in section 5, the basic equations (71) through (73).

The algorithm is based on the direct iteration technique, presented in this report, based on equations (71), (72) and (73). Some other indirect use of these equations have been tried and proved to offer some advantages, and are not discussed here.

A simple modification to the computer program is given in section C.3-a, which provides for a second sweep, in the reverse direction, when updating the vorticity vector W(I) using the point under-relaxation technique. This second sweep proved to enhance the convergence trend in some cases.

In the following computer program all geometric coefficients matrices are computed once and stored as a part of the program. This would not be possible for large matrices, in which case these matrices should be either computed element by element whenever needed, or stored on a desk mass storage and read into the computer memory one by one, or row by row, when needed. The later technique was used to generate the presented results for the example of flow inside a constricting channel.

C.2 List of Fortran Symbols

Fortran name	Description
AE(I)	Geometric parameters, associated with triangular elements. Equations (79) and (82)
BE(I)	Geometric parameters, associated with triangular elements. Equations (80) and (82)
BV(I)	An array used to store the right-hand side of a system of linear algebraic equations. Input to subroutine SUBST.
CE(I)	Geometric parameters, associated with triangular elements. Equations (81) and (82)
EPS	The convergence criterion parameter ϵ, equation (101).
ER	The maximum relative difference in ω between two successive iterations as determined by the left-hand side of equation (101)
HB(I)	An array containing the quantities $\mathrm{Re}\left(\frac{V^2}{2} + \frac{P}{\rho}\right)$
HBF	An algebraic parameter, equation (97)
HBM(I, J)	An array containing the elements of the geometric-coefficient matrix [H], Equations (73) and (84).
INT	(NB+1), the index of the first interior point. The first NB points are the boundary nodes.
IPVTH(I)	An array containing the pivoting strategy for the factored matrix HBM (I, J)
IPVYW(I)	An array containing the pivoting strategy for the factored matrix SMW(I, J).
NB	The number of grid points located on the boundary

NFE	The number of elements in the finite-element mapping of the region R.
NIT	The allowed maximum number of iterations
NBM1	(NB-1), the M character, generally, designates 'subtraction'
NE(I, J)	An array containing the indices of the three nodes (J = 1,2,3) of each element in the finite element mapping (I = 1, 2,..., NFE)
REN	The Reynolds number
SF(I)	The integrals $I^{\Delta}_{m,n}$, computed in Subroutine GF.
SMH(I, J)	An array containing the elements of the matrix HBM(I, J), after being factored by the Gauss elimination procedure. The factorization is done by the Subroutine SVB.
SMW(I, J)	An array containing the elements of the matrix [S], eqns.(88,89), after being factored by the Gauss elimination procedure. The factorization is done by Subroutine SVB
TB(I, J)	An array containing the x-component, TB(I, 1), and the y-component, TB(I, 2), of the unit tangent to the boundary B.
U(I)	The x-velocity components
UB(I)	The vector {UB}, equations (69) and (71). Later in the program, the first NB locations are used to store the vector $\{t_x u_b + t_y v_b\} - \{T\}$ equation (95)
UR	The under-relaxation parameter ur, equation (96).
URB	(1-UR)
V(I)	The y-velocity components
VB(I)	The vector {VB}, equations (70) and (72)
W(I)	The vorticity at the grid nodes

WBM(I, J)	An array containing the elements of the matrix [G], equations (73) and (83).
WU(I)	W(I) * U(I)
WV(I)	W(I) * V(I)
W1M(I, J)	An array containing the elements of the geometric coefficients matrix [D], equations (71) and (76).
W2M(I, J)	An array containing the elements of the matrix [E], equations (72) and (77)
X(I)	The x-coordinates of the grid nodes.
XV(I)	An array used to store the solution vector of a system of linear algebraic-equations. An output of Subroutine SUBST.
Y(I)	The y-coordinates of the grid nodes.

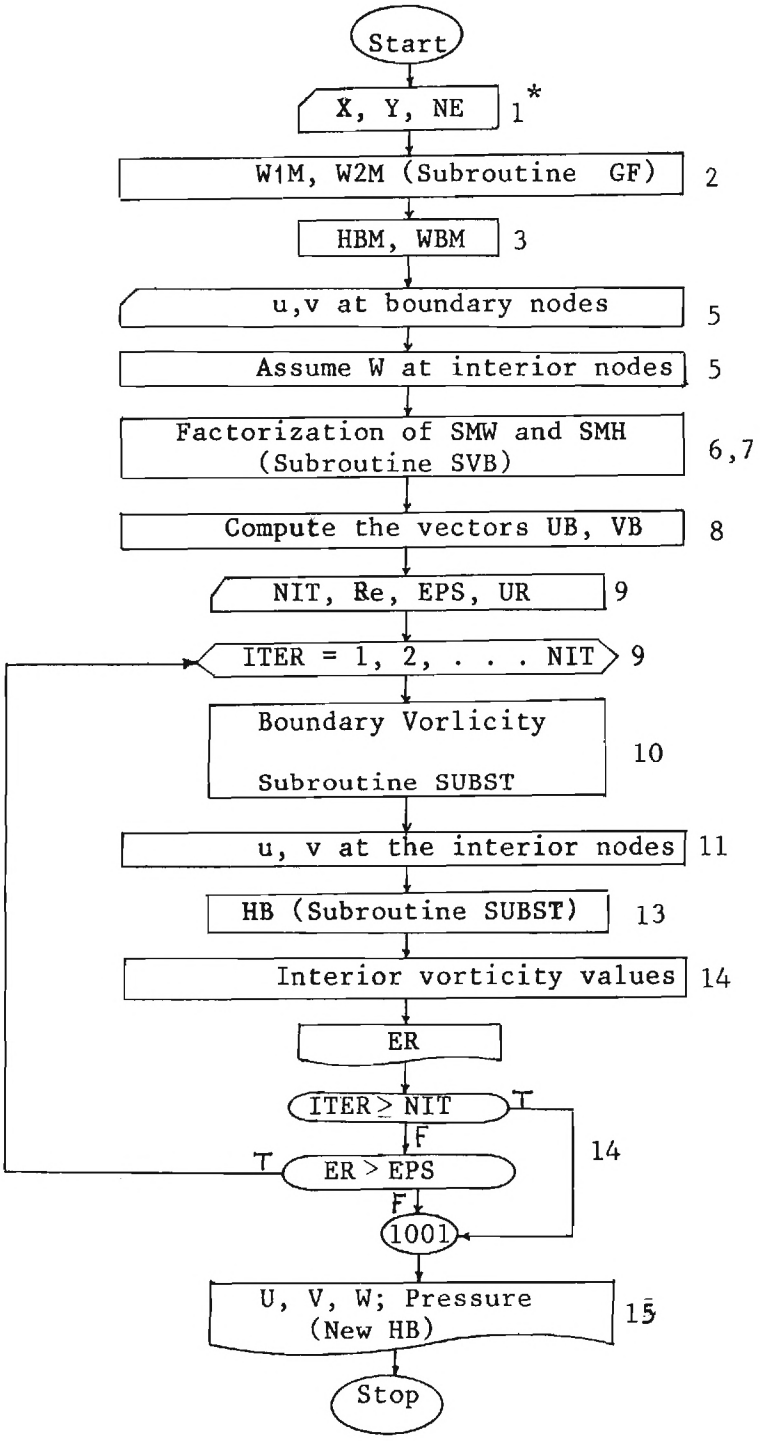


Figure C.1 Flow-Diaagram of the Main Program

* Numbers to the right of each block refer to the corresponding section in the computer code

C.3 Program Listing

```
      PROGRAM MAIN(INPUT,OUTPUT,TAPE5=INPUT,TAPE6=OUTPUT)
      COMMON /J2/ XE(4),YE(4)
      DIMENSION AE(3),BE(3),CE(3),SF(5)
      DIMENSION ITN(9,9),X(81),Y(81),NE(128,3)
      DIMENSION W1M(81,81),W2M(81,81),HBM(81,32),WBM(81,32)
      DIMENSION SMW(32,32),SMH(32,32),IPVTW(32),IPVTH(32)
      DIMENSION J(81),V(81),UB(81),VB(81),W(81),HB(32)
      DIMENSION TB(32,2),BV(32),XV(32),WU(81),WV(81)

      PIE=3.1415926535898
      TPIF=.5/PIE
C************************************* (1) (X,Y,NE)
      READ(5,*)NT,NB,NFE
      READ(5,*)X
      READ(5,*)Y
      READ(5,*)NE
      WRITE(6,4)NT,NB,NFE
      WRITE(6,5)X
      WRITE(6,6)Y
      WRITE(6,7)NE
4     FORMAT
5     FORMAT
6     FORMAT
7     FORMAT

      INT=NB+1
      NBM1=NB-1
      NBM2=NB-2
```

```
C****************************************** (2) (W1M,W2M)
      DO 100 I=1,NT
      DO 100 J=1,NT
      W1M(I,J)=0.
      W2M(I,J)=0.
100   CONTINUE
      DO 110 IE=1,NFE
      DO 115 I=1,3
      K=NE(IE,I)
      XE(I)=X(K)
      YE(I)=Y(K)
115   CONTINUE
      XE(4)=XE(1)
      YE(4)=YE(1)
      AE(1)=XE(2)*YE(3)-XE(3)*YE(2)
      AE(2)=XE(3)*YE(1)-XE(1)*YE(3)
      AE(3)=XE(1)*YE(2)-XE(2)*YE(1)
      DET=AE(1)+AE(2)+AE(3)
      DEN=1./DET
      AREA=0.5*ABS(DET)
      AE(1)=AE(1)*DEN
      AE(2)=AE(2)*DEN
      AE(3)=AE(3)*DEN
      BE(1)=(YE(2)-YE(3))*DEN
      BE(2)=(YE(3)-YE(1))*DEN
      BE(3)=(YE(1)-YE(2))*DEN
      CE(1)=(XE(3)-XE(2))*DEN
      CE(2)=(XE(1)-XE(3))*DEN
      CE(3)=(XE(2)-XE(1))*DEN
      DO 120 IP=1,NT
      XO=X(IP)
      YO=Y(IP)
      CALL SF(XO,YO,SF)
      SF(5)=AREA-SF(4)
      B1=XO*SF(2)+SF(3)
      C1=YO*SF(2)+SF(4)
      B2=XO*SF(1)+SF(5)
      C2=YO*SF(1)+SF(3)
      DO 125 J=1,3
      K=NE(IE,J)
      W1M(IP,K)=W1M(IP,K)+AE(J)*SF(2)+BE(J)*B1+CE(J)*C1
      W2M(IP,K)=W2M(IP,K)+AE(J)*SF(1)+BE(J)*B2+CE(J)*C2
125   CONTINUE
120   CONTINUE
110   CONTINUE
```

```
C****************************************** (3) (HBM,WBM)
      DO 150 I=1,NT
      DO 150 J=1,NB
      HBM(I,J)=0.
      WBM(I,J)=0.
150   CONTINUE

      DO 160 IB=1,NB
      I=IB
      J=I+1
      IF( IB .EQ. NB ) J=1
      DX=X(J)-X(I)
      DY=Y(J)-Y(I)
      SL=SQRT(DX*DX+DY*DY)
      DXOL=DX/SL
      DYOL=DY/SL

      DO 165 IO=1,NT
      IF( IO .EQ. I .OR. IO .EQ. J ) GO TO 165
      XOI=X(IO)-X(I)
      YOI=Y(IO)-Y(I)
      XC=(XOI*DXOL+YOI*DYOL)
      YC=(XOI*DYOL-YOI*DXOL)
      SMX=SL-XC
      YCS=YC*YC
      F1=ATAN2(SL*YC,YCS-SMX*XC)
      F2=0.5*ALOG((SMX*SMX+YCS)/(XC*XC+YCS))
      YOL=YC/SL
      XOL=XC/SL
      AJ=1.-YOL*F1+XOL*F2
      HBM(IO,I)=HBM(IO,I)+AJ-F2
      HBM(IO,J)=HBM(IO,J)-AJ
      AJ=YOL*F2+XOL*F1
      WBM(IO,I)=WBM(IO,I)+AJ-F1
      WBM(IO,J)=WBM(IO,J)-AJ
165   CONTINUE
160   CONTINUE

      DO 190 IB=1,NB
      IA=IB-1
      IF( IB .EQ. 1 )IA=NB
      IC=IB+1
      IF( IB .EQ. NB ) IC=1
      XI=X(IB)-X(IA)
      YI=Y(IB)-Y(IA)
      XJ=X(IC)-X(IB)
      YJ=Y(IC)-Y(IB)
      RIS=XI*XI+YI*YI
      RJS=XJ*XJ+YJ*YJ
      CIJ=(XI*XJ+YI*YJ)/SQRT(RIS*RJS)
      IF( CIJ .GT. 1. ) CIJ=1.
      IF( CIJ .LT. -1. )CIJ=-1.
      B=ACOS(CIJ)
      IF( (XI*YJ-YI*XJ) .LT. 0. ) B=-B
      WBM(IB,IB)=WBM(IB,IB)+PIE+B
      HBM(IB,IA)=HBM(IB,IA)+1.
      HBM(IB,IB)=HBM(IB,IB)-.5*ALOG(RJS/RIS)
      HBM(IB,IC)=HBM(IB,IC)-1.
190   CONTINUE
```

```
C**************************************** (4)
      DO 195 I=1,NT
      DO 195 J=1,NT
      W1M(I,J)=TPIF*W1M(I,J)
      W2M(I,J)=TPIF*W2M(I,J)
195   CONTINUE
      DO 196 I=1,NT
      DO 196 J=1,NB
      HBM(I,J)=TPIF*HBM(I,J)
      WBM(I,J)=TPIF*WBM(I,J)
196   CONTINUE

C**************************************** (5) (BC,IC)

      READ(5,*)(U(I),I=1,NB)
      READ(5,*)(V(I),I=1,NB)

      DO  360  I=1,NT

      W(I)=0.

360   CONTINUE
```

```
C****************************************** (6) (SMW)
      DO 400 IB=1,NB
      IA=IB-1
      IF( IB .EQ. 1 ) IA=NB
      IC=IB+1
      IF( IB .EQ. NB ) IC=1
      XI=X(IB)-X(IA)
      YI=Y(IB)-Y(IA)
      XJ=X(IC)-X(IB)
      YJ=Y(IC)-Y(IB)
      RI=SQRT(XI*XI+YI*YI)
      RJ=SQRT(XJ*XJ+YJ*YJ)
      XI=XI/RI
      YI=YI/RI
      XJ=XJ/RJ
      YJ=YJ/RJ
      TB(IB,1)=.5*(XI+XJ)
      TB(IB,2)=.5*(YI+YJ)
400   CONTINUE

      DO 410 I=1,NB
      DO 410 J=1,NB
      SMW(I,J)=TB(I,1)*W1M(I,J)-TB(I,2)*W2M(I,J)
410   CONTINUE
      CALL SVB(SMW,IPVTW,NB,NB)


C****************************************** (7) (SMH)
      DO 420 I=1,NB
      DO 420 J=1,NB
      SMH(I,J)=HBM(I,J)
420   CONTINUE

      SL1=SQRT((X(NB)-X(NBM1))**2+(Y(NB)-Y(NBM1))**2)
      SL2=SQRT((X(NBM1)-X(NBM2))**2+(Y(NBM1)-Y(NBM2))**2)
      HBF=SL1/(SL1+SL2)
      DO 430 I=1,NB
      SMH(I,NBM2)=SMH(I,NBM2)+HBF*SMH(I,NBM1)
430   CONTINUE
      CALL SVB(SMH,IPVTH,NB,NBM2)
```

```
C**************************************** (8) (UB,VB)
      DO 450 I=1,NT
      UB(I)=0.
      VB(I)=0.
      DO 450 J=1,NB
      UB(I)=UB(I)+WBM(I,J)*U(J)+HBM(I,J)*V(J)
      VB(I)=VB(I)-HBM(I,J)*U(J)+WBM(I,J)*V(J)
450   CONTINUE

      DO 460 I=1,NB
      UB(I)=TB(I,1)*(U(I)-UB(I))+TB(I,2)*(V(I)-VB(I))
460   CONTINUE


C**************************************** (9) (ITER)**********************
      ITER=0
1001  READ(5,*)NIT,REN,EPS,URF
      IF(EOF(5))1002,1003
1003  WRITE(6,980)NIT,REN,EPS,URF
      URB=1.-URF

      HB(NB)=.5*REN*(U(NB)*U(NB)+V(NB)*V(NB))
      VS1=U(NBM1)*U(NBM1)+V(NBM1)*V(NBM1)
      VS2=U(NBM2)*U(NBM2)+V(NBM2)*V(NBM2)
      VSF=.5*REN*(VS1-HBF*VS2)

1000  ITER=ITER+1

C**************************************** (10) (WB)
      DO 500 I=1,NB
      W1=0.
      W2=0.
      DO 510 J=INT,NT
      W1=W1+W1M(I,J)*W(J)
      W2=W2+W2M(I,J)*W(J)
510   CONTINUE
      BV(I)=UB(I)-TB(I,1)*W1+TB(I,2)*W2
500   CONTINUE
      CALL SUBST(SMW,BV,XV,IPVTW,NB)
      DO 520 I=1,NB
      W(I)=XV(I)
520   CONTINUE

C**************************************** (11) (VIN)
      DO 550 I=INT,NT
      U1=0.
      V1=0.
      DO 560 J=1,NT
      U1=U1+W1M(I,J)*W(J)
      V1=V1+W2M(I,J)*W(J)
560   CONTINUE
      U(I)=UB(I)+U1
      V(I)=VB(I)-V1
550   CONTINUE
```

```
C**************************************** (12) (WU,WV)
      DO 600 I=1,NT
      WU(I)=W(I)*U(I)
      WV(I)=W(I)*V(I)
600   CONTINUE

C**************************************** (13) (HB)
      DO 620 I=1,NB
      W1=0.
      DO 630 J=1,NT
      W1=W1+W2M(I,J)*WU(J)+W1M(I,J)*WV(J)
630   CONTINUE
      W3=0.
      DO 632 J=1,NB
      W3=W3+WBM(I,J)*W(J)
632   CONTINUE
      BV(I)=-W(I)-REN*W1+W3
      BV(I)=BV(I)-HBM(I,NBM1)*VSF-HBM(I,NB)*HB(NB)
620   CONTINUE
      CALL SUBST(SMH,BV,XV,IPVTH,NBM2)
      DO 640 I=1,NBM2
      HB(I)=XV(I)
640   CONTINUE

      HB(NBM1)=VSF+HBF*HB(NBM2)

C**************************************** (14) (WIN)
      DMAX=0.
      WMAX=0.
      DO 710 I=INT,NT
      W1=0.
      DO 720 J=1,NT
      W1=W1+W2M(I,J)*WU(J)+W1M(I,J)*WV(J)
720   CONTINUE
      W23=0.
      DO 725 J=1,NB
      W23=W23-HBM(I,J)*HB(J)+WBM(I,J)*W(J)
725   CONTINUE
      WN=-REN*W1+W23
      WN=URF*WN+UR8*W(I)
      DIF=W(I)-WN
      WMAX=AMAX1(WMAX,ABS(WN))
      DMAX=AMAX1(DMAX,ABS(DIF))
      W(I)=WN
      WU(I)=W(I)*U(I)
      WV(I)=W(I)*V(I)
710   CONTINUE

      ER=DMAX/WMAX
      WRITE(6,983)ITER,ER

      IF( ITER .GE. NIT ) GO TO 1001
      IF ( ER .GT. EPS ) GO TO 1000
      GO TO 1001
```

```
C**************************************** (15)
1002  CONTINUE
      IF(REN .EQ. 0. )GO TO 1005
      DO 750 I=1,NB
      VS=U(I)*U(I)+V(I)*V(I)
      HB(I)=HB(I)/REN-.5*VS
750   CONTINUE

1005  CONTINUE
      WRITE(6,990)HB
      WRITE(6,991)U
      WRITE(6,991)V
      WRITE(6,991)W

974   FORMAT(/1X,4F10.5/)
980   FORMAT(//1X,I7,3F12.5//)
983   FORMAT(1X,I6,2F10.6)
990   FORMAT(
991   FORMAT(
      STOP
      END
```

```
C**************************************** (16) (SUB. GF )
      SUBROUTINE GF(XO,YO,SF)
      COMMON /J2/ XE(4),YE(4)
      DIMENSION SF(3)
      DELTA=1DE-4
      DO 1 I=1,4
      SF(I)=0.
1     CONTINUE
      DO 10 I=1,3
      J=I+1
      XI=XE(I)-XO
      YI=YE(I)-YO
      XJ=XE(J)-XO
      YJ=YE(J)-YO
      DX=XI-XJ
      DY=YI-YJ
      IF (ABS(DX) .GT. DELTA ) GO TO 30
      IF (ABS(XI) .LE. DELTA ) GO TO 10
      GO TO 50
30    PI=DY/DX
      CY=YI-PI*XI
      IF (ABS(CY) .LE. DELTA ) GO TO 10
      IF (ABS(DY) .LE. DELTA ) GO TO 60
      CX=-CY/PI
      A=CX/SQRT(CX*CX+CY*CY)
      D=CY*A
      B=-PI*A
      GO TO 100
50    D=XI
      A=0.
      B=1.
      GO TO 100
60    D=YI
      A=1.
      B=0.
100   RIS=XI*XI+YI*YI
      RJS=XJ*XJ+YJ*YJ
      COTI=(XI*A-YI*B)/(YI*A+XI*B)
      COTJ=(XJ*A-YJ*B)/(YJ*A+XJ*B)
      CIJ=(XI*XJ+YI*YJ)/SQRT(RIS*RJS)
      IF(CIJ .GT. 1.)CIJ=1.
      IF(CIJ .LT. -1.)CIJ=-1.
      F1=ACOS(CIJ)
      IF((XI*YJ-YI*XJ) .LT. 0.)F1=-F1
      F2=0.5*ALOG(RIS/RJS)
      F3=-(COTJ-COTI+F1)
      FF=-F3+F1
      D2=0.5*D*D
      AB=A*B
      A2=A*A
      B2=1.-A2
      SF(1)=SF(1)+D*(A*F2+B*F1)
      SF(2)=SF(2)+D*(A*F1-B*F2)
      SF(3)=SF(3)+D2*((A2-B2)*F2+AB*FF)
      SF(4)=SF(4)+D2*(F1-B2*FF-2.*AB*F2)
10    CONTINUE
      RETURN
      END
```

```
C**************************************** (17) (SUB. SVB )
      SUBROUTINE SVB(W,IPVT,NE,NU)
      DIMENSION W(32,32),IPVT(32),D(32)
      DO 2 I=1,NE
      IPVT(I)=I
      RMX=0.0
      DO 4 J=1,NU
      RMX=AMAX1(RMX,ABS(W(I,J)))
4     CONTINUE
      D(I)=RMX
2     CONTINUE
      NUM1=NU-1
      IF( NE .GT. NU )NUM1=NU
      DO 10 K=1,NUM1
      J=K
      IP=IPVT(K)
      CMX=ABS(W(IP,K))/D(IP)
      KP1=K+1
      DO 12 I=KP1,NE
      IP=IPVT(I)
      WOD=ABS(W(IP,K))/D(IP)
      IF (WOD .LE. CMX ) GO TO 12
      CMX=WOD
      J=I
12    CONTINUE
      IPK=IPVT(J)
      IPVT(J)=IPVT(K)
      IPVT(K)=IPK
      IF( K .EQ. NU ) GO TO 10
      DO 14 I=KP1,NE
      IP=IPVT(I)
      R=-W(IP,K)/W(IPK,K)
      W(IP,K)=-R
      DO 16 J=KP1,NU
      W(IP,J)=W(IP,J)+R*W(IPK,J)
16    CONTINUE
14    CONTINUE
10    CONTINUE
      RETURN
      END
```

```
C**************************************** (18) (SUB. SUBST )
      SUBROUTINE SUBST( W,B,X,IPVT,N)
      DIMENSION W(32,32),B(32),X(32),IPVT(32)
      IP=IPVT(1)
      X(1)=B(IP)
      DO 15 K=2,N
      IP=IPVT(K)
      KM1=K-1
      SUM=0.
      DO 14 J=1,KM1
      SUM=SUM+W(IP,J)*X(J)
14    CONTINUE
      X(K)=B(IP)-SUM
15    CONTINUE
      X(N)=X(N)/W(IP,N)
      K=N
      DO 20 NK=2,N
      KP1=K
      K=K-1
      IP=IPVT(K)
      SUM=0.
      DO 19 J=KP1,N
      SUM=SUM+W(IP,J)*X(J)
19    CONTINUE
      X(K)=(X(K)-SUM)/W(IP,K)
20    CONTINUE
      RETURN
      END
```

C. 3-a Program Modification*

```
      NI=NT-NB
      DO 730 K=1,NI
      I=NT+1-K
      W1=0.
      DO 740 J=1,NT
      W1=W1+W2M(I,J)*WU(J)+W1M(I,J)*WV(J)
740   CONTINUE

      W23=0.
      DO 745 J=1,NB
      W23=W23-HBM(I,J)*HB(J)+WBM(I,J)*W(J)
745   CONTINUE
      WN=-REN*W1+W23
      W(I)=URF*WN+URB*W(I)
      WU(I)=W(I)*U(I)
      WV(I)=W(I)*V(I)
730   CONTINUE
```

* This set of cards should be inserted, if desired, after the card (710 CONTINUE) to effect a second sweep, in the reverse direction, when updating the array W(I) , as explained earlier in section C.1.

References

1. Wu, J. C. and Thompson, J. F., "Numerical Solutions of Time-Dependent Incompressible Navier-Stokes Equations Using an Integro-Differential Formulation," Computers and Fluids, 1, pp. 197-215, 1973.

2. Wu, J. C., "Integral Representations of Field Variables for the Finite Element Solution of Viscous Flow Problems," Proc. of the 1974 Conference on Finite Element Methods in Engineering, University of New South Wales Press, Sydney, N. S. W., Australia, pp. 827-840, 1974.

3. Wu, J. C., "Finite Element Solution of Flow Problems Using Integral Representations," Proceedings of the 2nd International Symposium on Finite Element Methods in Flow Problems, International Center for Computer Aided Design, Conf. Series No. 2, pp. 205-216, 1976.

4. Wu, J. C., and Wahbah, M. M., "Numerical Solution of Viscous Flow Equations Using Integral Representations," Lecture Series in Physics, 59: Proceedings of the Fifth International Conference on Numerical Methods in Fluid Dynamics, pp. 448-453, Springer-Verlag, 1976.

5. Wu, J. C., "Numerical Boundary Conditions for Viscous Flow Problems," AIAA Journal, 14, pp. 1042-1049, 1976.

6. Wu, J. C., Wahbah, M. M., and Sugavanam, A., "Some Numerical Solutions of Turbulent Flow Problems by the Use of Integral Representations," Proceedings of the Symposium on Applications of Computer Methods in Engineering, University of Southern California, Los Angeles, California, 1977.

7. Wahbah, M. M., "Evaluation of Shape Functions Associated with the Finite Element Solution of the Integral Representations for Two-dimensional Flows," Unpublished Notes, 1976.

8. Taylor, R. L., "On Completeness of Shape Functions for Finite Element Analysis," Int. J. Num. Meth. Engng., 4, pp. 17-22, 1972.

9. Conte, S. D., and De Boor, C., "Elementary Numerical Analysis," McGraw-Hill Book Co., N. Y., pp. 127-133.

10. Burggraf, O. R., "Analytical and Numerical Studies of the Structures of Steady Separated Flows," J. Fluid Mech., 24, pp. 113-151, 1966.

11. Runchal, A. K., Spalding, D. B., and Wolfshtein, "Numerical Solution of the Elliptic Equations for Transport of Vorticity, Heat, and Matter in Two-Dimensional Flow," Phys. Fluid, 12, pp. II-21-II-28, 1969.

12. Taylor, C., and Hood, P., "A Numerical Solution of the Navier-Stokes Equations Using the Finite Element Technique," Computers and Fluids, 1, pp. 73-100, 1973.

13. Cheng, R. T., "Numerical Solution of the Navier-Stokes Equations by the Finite Element Method," Phys. fluids, 15, pp. 2098-2105, 1972.

14. Rektoys, K., "Survey of Applicable Mathematics," MIT Press, p. 549, 1969.

15. Launder, B. E. and Spalding, D. B., "The Numerical Computation of Turbulent Flows," Computer Methods in Applied Mechanics and Engineering, 3, pp. 269-289, 1974.

16. Laufer, J., "Investigation of Turbulent Flow in a Two-Dimensional Channel," NACA Report 1053, 1951.